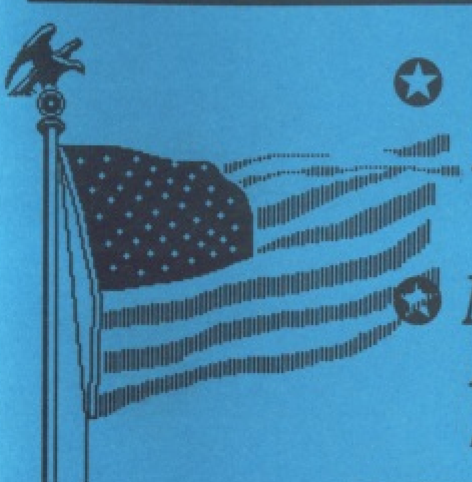
Page 6 Publishing's **NEW**

ATARI USER

The Resource for the ATARI CLASSIC and the ATARI ST

Issue 77 · July/August 1996 £2.50

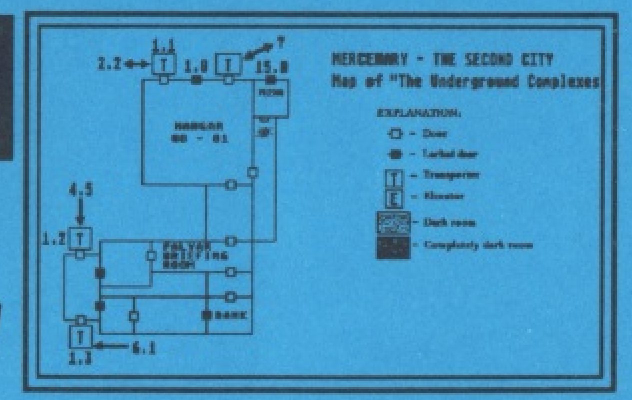
FOR THE ATARI CLASSIC



- ## *SOFTWARE ISSUE*
  Reviews of half a dozen recent software releases
- ## *MACHINE CODE GRAPHICS*
  It ain't easy but we show you how it can be done
- ## *HIGH SCORE TABLE*
  Add the professional touch

THE TIPSTER

brings you incredible maps of MERCENARY!



PLUS ... TURBO TIPS ... TEXTPRO MACROS ... CHARACTER SET REDEFINER ... AND MUCH MORE!

This issue's ....

Thanks

Les Ellingham puts it all together and fills up the gaps but the real thanks goes to the following who made this issue possible

Sandy Ellingham who takes care of all the office work, advertising and mail order

For their regular contributions

John S Davison **Allan J. Palmer**
Paul Rixon **Stuart Murray**
Ann O'Driscoll

For their contributions this issue

Ralph Espino **Bill Halsall**
Avram Dumitrescu **Andy Guillaume**
Daniel Baverstock **Roald Sund**
John Foskett **Frank Walters**
Peter Foote **H S Wood**

APOLOGIES

I am still extremely poor in acknowledging contributions so I apologise to everyone who has sent in stuff and thought it has gone through the wormhole. The intention to reply to everyone is there but the time seems to drift by. If you have not heard, thank you and keep watching the mag, you might be surprised.

HOW IT'S DONE

PAGE 6 shows just what you can do with your Atari. NEW ATARI USER has always been created entirely with Atari equipment, initially on the XL but more lately with a Mega ST and other stuff, who needs PC's or Macs! Hardware includes a Mega ST2 (upgraded to 4Mb), SM125 Monitor, Supra 30Mb Hard Disk, a HP Laserjet III, Citizen 124D printer, Philips CM8833 monitor, 130XE, a couple of 1050 disk drives, 850 interface, NEC 8023 printer. Principal software used is Protext and Fleet Street Publisher 3.0. Other software includes Kermit, TariTalk, Turbo Basic and various custom written programs on the XL/XE. Articles submitted on XL/XE disks are transferred across to the ST via TARITALK. Programs are coded on the XE and printed out directly for pasting in after the typesetting is completed. All major editing is done with Protext and pages are laid out with Fleet Street Publisher. Each page is output directly from Fleet Street to a HP Laserjet III which produces finished pages exactly as you see them. All that is left is to drop in the listings and photos.

Well, it's not quite as easy as that but you get the idea!

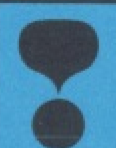
Inspiration

It's Kathy Mattea once again, all the way through this issue although The Incredible String Band did get one listen. Those that I would like to have had on the platter were Neil Young who looks to have another good one just out and Mark Knopfler whose solo effort I listened to in Woolworths and it sounded rather good. Early on I thought the music had had it for good as the CD just sort of died, but like New Atari User it's made of sterner stuff and came back to life a few days later

CONTRIBUTIONS

Without contributions from its readers, NEW ATARI USER would not be possible. PAGE 6 welcomes and encourages its readers to submit, articles, programs and reviews for publication. Programs must be submitted on disk or cassette, articles should wherever possible be submitted as text files on disk. We seek to encourage your participation and do not have strict rules for submissions. If something interests you, write a program or article and submit it!

COPYRIGHT



Editorial address: P.O. Box 54, Stafford, ST16 1DR, ENGLAND Tel. 01785 241153
Editor & Publisher: Les Ellingham - *Advertising:* Sandy Ellingham
Page layout by PAGE 6 - Printed by Dolphin Press, Fife, Scotland 01592 771652
NEW ATARI USER is published bi-monthly on the last Thursday of the month prior to cover date

PAGE 6 PUBLISHING's
NEW
ATARI
USER

'The Magazine for the Dedicated Atari User'

ISSN No. 0958-7705

CONTENTS

Issue 77 - July/August 1996

REGULARS

ADVERTISERS

PROGRAMMING

FEATURES

REVIEWS

SUBSCRIPTIONS

MAGAZINE ONLY

Annual subscription rates (6 issues)

UK	£15.00
Europe (Air Mail)	£17.00
Elsewhere (Surface)	£17.00
Elsewhere (Air Mail)	£23.00

Overseas rates reflect only the difference in postal costs

DISK SUBSCRIPTION

A disk containing all of the 8-bit programs from each issue of NEW ATARI USER is available either separately or on subscription. Single price £2.95 per disk, a disk subscription saves you almost £8 a year. Subscription rates (6 issues)

UK	£25.00
Europe	£32.00
Elsewhere (sea)	£32.00
Elsewhere (Air)	£42.00

Please make cheques payable to PAGE 6 PUBLISHING and send to PAGE 6 Publishing, P.O. Box 54, Stafford, ST16 1DR

Editorial

It was great this time to receive not one, but three, quality programs that we can use for the Disk Bonus so thanks to Bill Halsall, Robert de Letter and Graeme Fenwick who will tide us over for the next few issues. We do have a number of articles in reserve but, as always, rely entirely on you to keep the ideas and articles coming. Even if you have submitted things in the past, keep up the momentum and let us have a new article or program for the coming issues.

In the Mailbag column Brad Rogers puts forward the idea that the mag be put on a quarterly basis because we have been a bit late with recent issues. The reasons for the delays have been touched on in previous issues and, overall, I think that we are doing quite well to keep the mag going given all the pressures and difficulties of making a living in Mr Major's 'most successful European economy'.

One of the advantages of a bimonthly schedule if that it keeps the impetus going to keep producing regular issues and even if we fall behind from time to time it's generally not disastrous. If we officially go to a quarterly basis the temptation might be there to slip behind on that schedule as well because we all know that no matter how much time there is to do something, it is always done at the last minute! With a bimonthly schedule you will get at least five copies in a twelve month period and that is better than four, or even three! Talking of the last minute, you will find that several regular columns are missing from this issue because I waited until the last minute but they didn't arrive. We had a sudden deadline forced upon us by the printers announcing their annual holiday enclosure and we had to get the mag to them by a certain date or wait an extra three weeks for it to be printed. I know that the columns were posted to us for the deadline but you still cannot rely on Royal Mail, even though they put their prices up in a few days time. Whereas we would normally hold on for a day or so to allow the post to arrive I thought it more important this time to make sure that the mag was printed without the three weeks delay. See, I do try to get it out on time!

The thing to do is to stick with us, understand that there may be delays once in a while and appreciate that you still have a magazine that supports the Atari Classic. We have had many letters and comments of support and they are all gratefully received.

Les Ellingham

TURBO BASIC

BACK TO DOS

Ralph Espino solves one of the real bugbears for Turbo users - how to get back to Turbo once you've called DOS

This short program modifies your copy of DOS 2.5 so as to allow you to easily return to TURBO BASIC after entering DOS without re-booting.

To use it, first boot up with TURBO BASIC, type in the program and save a copy before running it. Then run the program and, when prompted to do so, insert a disk containing the copy of DOS 2.5 that you want to modify and press RETURN. After a short wait you will then be transported to DOS. Once in DOS, select the L option (LOAD FILE) and enter the filename ENHDOS. Your version of DOS should now be modified. This can be confirmed by pressing RETURN. The B. (RUN CARTRIDGE) line should now read RUN TURBO BASIC. Finally you must save the modified version to disk. This is done by using the H option (WRITE DOS FILES) from the DOS menu.

Now whenever you are in DOS and want to return to TURBO BASIC all you need do is press B and RETURN, and there you have it. There are two things to watch out for though, firstly this new version of DOS can only be used to return to TURBO BASIC and NOT to ATARI BASIC or any other language so you should only modify the DOS's on your TURBO BASIC disks. And secondly you need to have the MEM.SAV file on the disk at all times, if you use any commands that invalidate MEM.SAV, i.e. DUPLICATE FILE, then you will not be able to return to TURBO BASIC. ●

```
IL 1 REM ##############################
OI 2 REM #   DOS ENHANCER FOR TURBO   #
NT 3 REM #            by              #
PY 4 REM #        Ralph Espino        #
XR 5 REM # -------------------------- #
YU 6 REM # NEW ATARI USER - JULY 96   #
IR 7 REM ##############################
NN 8 REM
UI 10 ? "[ESC,CLEAR]INSERT DISK AND PRESS
   RETURN ":GET K:? :? :? "PLEASE WAIT.
   "
ZX 20 CLOSE :OPEN #%2,8,%0,"D:ENHDOS":FOR
    Q=%1 TO 81:READ A:PUT #%2,A:NEXT Q:CL
   OSE
EZ 30 OPEN #%2,8,%0,"D:MEM.SAV":BPUT #%2,
   8192,5625:CLOSE :DOS
EE 40 DATA 255,255,0,120,68,120,169,240,1
   41,91,39,169,234,141,119,39,141,120,39
   ,169,76,141,29,25,169,128,141
DT 50 DATA 30,25,169,32,141,31,25,169,54,
   133,204,169,120,133,205,160,14,169,119
   ,133,206,169,31,133,207,177,204
BR 60 DATA 145,206,136,208,249,96,82,85,7
   8,32,84,85,82,66,79,32,66,65,83,73,67,
   224,2,225,2,0,120
```

Underline = INVERSE CHARACTERS · [ ] = CONTROL + CHARACTER · < > = INVERSE CONTROL + CHARACTER

Mailbag



Oh, no! Not again!

If you are looking forward to Allan Palmer's Mailbag this issue then sorry! Once again it didn't quite work out as by the time we got enough letters for the column it was too late to send them on. You are stuck with me again this issue. Please try to write as soon as you can after you have read this column so that we can try and get Allan back on board.

Les Ellingham

MACHINE CODE GRAPHICS

It is great when problems posed in one issue are answered by the next and we must thank David Sargeant from Grantham for his help in solving a problem posed last issue. David writes: "In the last issue there is a letter from James Mathrick asking how to change graphics mode in machine code which I would like to answer.

You say that you have played around with location 623. Maybe you are looking for a simple POKE command that would change the mode for you, but as far as I now this can't be done. In BASIC you print to a graphics screen using the PRINT #6 command and this should give you an idea of how to solve your problem. You set up an Input/Output Control Block in the same way as BASIC does when you use a graphics command such as GRAPHICS 3. IOCB's have been covered in past issues of NAU, so I will not go into any detail here. They are usually used to interface with disk drives, printers or the keyboard, but one can also be assigned to the screen.

This is how I would write a routine which could be called from BASIC with the USR command.

```
        ORG 1536
GRAPHS  PLA        ; Number of values
        LDX #16    ; Load X with offset
                     to IOCB #1
        LDA #3     ; Store Open com-
                     mand in ICCOM
        STA 834,X  ; Store low and high
                     bytes
        LDA #32    ; of DEVICE address
        STA 836,X  ; in ICBAL/H
        LDA #6
        STA 837,X
        LDA #24    ; Store screen type in
                     ICAX1
        STA 842,X
        LDA #3     ; Store graphics mode
                     in ICAX2
        STA 843,X
        JSR 58454  ; Jump to Central
                     Input/Output Vector
        RTS        ; Return to BASIC
DEVICE  DB 83,58   ; ASCII of screen
                     device S:
```

The machine produced is 104,162,16,169,3,157,66,3, 169,32,157,68,3,169,6,157, 69,3,169,24,157,74,3,169,3, 157,75,3,32,86,228,96,83,58

I have checked this routine to make sure it was correct before sending it to you and hope it works alright. If you print it in the next issue of NAU, I hope it will be helpful to James in his programming."

❢ *Again many thanks, David. We are more than happy to publish you letter which will no doubt help others besides James. This is what Mailbag is all about.*

XIO HINT

A little hint here from Peter Foote in Yeovil which you might find useful. "A quick hint, maybe a tip, regarding the use of BASIC graphics statements. A problem I recently had with the XIO command was that:
XIO 18, #6,0,0,"S:"
followed by
LOCATE X,Y,C
created Error 131 - IOCB Write Only.

To correct this use the command:
XIO 18, #6,4,0,"S:"

❢ *Many thanks. Anyone else got any handy little tips like this?*

MULTIPLE QUERIES

When sending in his renewal David Bennett from Coventry raised a number of queries which are of general interest so I thought that we would put his letter in Mailbag and maybe provide answers to some of the things other readers have been wondering about. There are quite a few questions so let's tackle them in order. David starts by asking "You show a telephone number - is this manned 24 hours or are there specific times when someone is there to answer queries, as opposed to an answering machine?"

❢ *Well, the phone is not exactly manned 24 hours a day, I do like to get some sleep sometimes! We no longer use the answering machine, except on rare occasions, so if you ring you will either get an answer or know we are out. The phone is really for routine enquiries and orders and is not intended to be a help line for problems although I am happy to help out if I can. All that I ask is that anyone phoning does so at a sensible time. I got rather fed up some years ago when people would phone at 11 pm on a Sunday expecting us to help them out.*

The next question is "Are there any PAGE 6 binders still available? I took the original Atari User magazines since the first issue and New Atari User since the original Atari User ceased. I have also taken other magazines on a less regular basis and have been getting ST Format for some time as it now seems to be the main magazine for the ST. My problem is that the ST User binders, like the old Atari User binders, have wire rods to hold the magazines in place and take up about twice the width needed for the magazines. The PAGE 6/ New Atari User binders had elastic to hold the magazines in place which made them slimmer, much better for my shelves. Naturally neither type shows a manufacturer and I cannot find any local stockists for this type of binder"

❢ *I have always hated the 'wire rod' binders as they are bulky and the magazines tend to be a bit floppy. Other companies use them though because they are cheaper than the type we used to use. Our binders were specifically designed for us to hold a dozen issues exactly of the magazine and were quite neat. The problem was that they were expensive and the last batch we ordered meant that we were selling them almost at cost by the time the postage was taken into account. Add to that a minimum order of about 400 binders and you can understand why we decided to discontinue them. This is also the reason why no binders are available for the new smaller size magazine, it simply wouldn't be cost effective for us to have them designed and made. They would have to sell at around £6/7 and there*



would be too few readers wanting them at that price. Some years ago there was a company that supplied 'one off' binders with the wire fixings which came with a set of transfer letters so that you could add your own title but I have not seen these advertised for many years. Does anyone know if these are still available?

The next query concerns the PD catalogues "Can you please confirm the latest issues of the Catalogues? I have a loose-leaf XL/XE catalogue dated 08/95, various update pages and a 1993 XL/XE catalogue which is presumably out of date and should be scrapped. The ST catalogue I have is dated 1992 - is this the latest or can I please have the latest issue?

❢ *The old style 'stapled' XL/XE catalogue is out of date and was replaced in 1995 by a new style loose-leaf catalogue which we intended to keep up to date by regularly issuing new pages. This idea was fine but we never quite worked out how to ensure that readers who purchased the catalogue got the updated pages. We should have kept a database of purchasers but kept forgetting. In theory we could send out updated pages now but everything changed with the famous hard disk crash which affected the last issue. The new style catalogue was one of the things that was lost and I had to recreate it from scratch. Unfortunately it turned out to be different in format than the original so that the pages are not the same. Any updated pages will have some disks overlapping, a bit messy but not too much of a problem. The best thing for anyone to do about getting updated pages for their XL/XE catalogue is to drop us a line, or phone, asking for updates and telling us the last disks (standard and DS) on their current pages. We can then mail out the new pages.*

As I am typing this it occurs to me that we could enclose an 'update request form' with each catalogue which readers could mail back every so often for their new pages. Sounds like that could work.

As far as the ST catalogue goes, we no longer print a full catalogue so hang on to what you have as it won't be replaced. All of the disks are still available and we can mail out individual updates although these are not in a form that fit in the catalogue. Again simply let us know what the last disk is in the catalogue you have and we will send details of any new disks added.

Next question is slightly harder to answer "All the adverts for machines, etc. seem to be from places that I can't get to as I no longer have a car. What is the best way of arranging safe collection or delivery?"

❢ *If you want to be sure of something arriving don't use Parcelforce! The last lot of disks we ordered were sent on a 48 hour guaranteed delivery with a bar-coded tracer system on the parcel. When they hadn't arrived 10 days later we asked Parcelforce what had happened only to be told that they had no trace of the parcel anywhere in their system. Royal Mail and Parcelforce are certainly the cheapest but be prepared for something to be lost sooner or later. The only other ways are by the regular carriers who generally are more reliable but you have to pay quite a lot for the service. Is there actually a service that is 100% reliable?*

Technical queries now with "I have an 800XL with a 1027 printer and an STE with a Citizen Swift print. Is it possible to connect both computers to a switch and just use the Citizen printer. If so could you advise me of a supplier for switch and leads and is it worth doing?"

❢ *Well, you can certainly get a switch box to connect two computers to a single printer. I don't know a supplier offhand but there are a number of companies that advertise in the PC magazines that could probably help. These boxes vary in price from around £5/6 to up to £50 or so, so it is worth looking around. The main problem is that you are going to need an interface for your 800XL. Although the XL works with the 1027, it cannot be connected directly to a Centronics printer (such as the Citizen) and you need a Centronics interface. You can get one of these for around £30 from Micro Discount. All in all it is likely to cost you around £50 to get connected, only you know if that is a price worth paying.*

And lastly - "A last point, mainly with using Shareware. Most things I see and would consider paying for seem to ask for payment in foreign currency which makes it expensive. There was some talk quite a time back of a clearing address in the UK that would accept payment in UK currency and forward payments. Is there still such an organisation and do you have any information about it?"

❢ *Shareware is a nightmare as regards payments. How do you know whether the guy asking for the donation is still supporting the software? How do you know that an 'intermediary' will pass on the money? As far as I know no-one has come up with a totally satisfactory way of handling shareware monies. It is probably best to treat the matter as a question of conscience and with the knowledge that you may never know whether your cash has arrived. Best thing is probably to decide whether you can risk the amount asked for and then pop down the bank, get some dollars or Deutschmarks, send them off and hope for the best!*

HARD DISKS ETC.

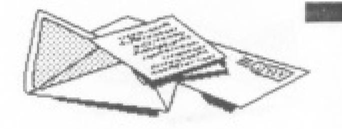
Regular correspondent Brad Rogers from Southampton tackles a number of points raised last issue taking Allan Palmer to task for the misinformation about hard disk drives, when in fact it was me (Les) who provided the spurious answer. Here's what Brad has to say - "In Issue 76 in connection with hard disk interfaces, you claimed that they are pretty much a dead duck. This is far from the truth in fact. There are currently three different systems available. Sadly they all originate from the USA so aren't particularly cheap.

First, the Supra system you mentioned is still available, albeit from a different manufacturer. Supra hold the rights. Secondly, there's the MIO board. Not only a HD interface, but it allows the addition of RAM too. Finally there is the Black Box which is similar in concept to the MIO. Of the three, probably the MIO is the most popular (it used to be manufactured by ICD) with the Black Box coming a close second. The Supra system is too limited, in that it can't accept additional drives, whereas the other two systems can. Sadly, I don't have pricing information available here at the moment, but it is readily available from the Internet. If there is sufficient interest, I could acquire details for publication.

Which leads me, quite nicely, to the second topic. Again, you indicated that around £1000 was required to 'get connected'. Simply put, this isn't true! I offer as evidence an article in the very same magazine (issue 76) by Gordon Hooper. In this article he explains what's required to get your Atari 8-bit on-line. Being on the 'net myself, I've seen evidence that it is done. There is plenty of Atari Classic software available, too.

Talking of software. One of the perennial arguments in comp.sys.atari.8-bit is





whether or not we (Atarians) should put copyright material up on the 'net, since 90% of the commercially produced material we have is no longer in production. You'll be pleased to hear that any attempts at this sort of piracy (I use the word advisedly) has so far failed to get anywhere. Sadly the same cannot be said of commercial C64 software. Nor, for that matter, PC software.

Now I have a question about the continued late arrival of NAU through my letter box. Realising Les, Sandy et al have to make a living, wouldn't it be better to slow the rate down? That is, turn the magazine into a quarterly, rather than the current (supposed) bimonthly? I have little doubt that there's still sufficient interest and material available to justify the bimonthly cover dates, but when the schedules are consistently not met, it makes you wonder doesn't it? Please don't misunderstand my intentions, I'm trying to make life easier for Les, not slag him off. Rest assured my subscription has been renewed."

❢ *This just the sort of letter we want for Mailbag, Brad. A few points answered, some comments and a few thought provoking ideas to mull, over. Let's talk about the hard disk matter first. I am delighted to be proved incorrect in my comments, as evidenced also by other letters received on this subject. Maybe it's time for someone to do a definitive article on just how far the 8-bit can be expanded and where the bits can be purchased and at what cost? As regards getting connected on the Internet I stand by what I said. I wasn't talking about folk who read New Atari User and other computer magazines who are 'into' computers already but the ordinary person who reads newspapers, buys magazines, watches the TV and maybe dabbles with Ceefax but does not already have a computer. How is he going to 'get connected'? Simple, he goes into his local Dixons, or Escom, or maybe PC World and asks for advice. 'Of course you can get on the Internet sir, we have a budget system here which starts at only £1,000'. Point made?*

The question of commercial software being made 'public domain' is very interesting and one which I would like a lot of feedback on in future Mailbag columns. The argument is long and complex but, on the whole, I tend to feel that the time has come to release all commercial software that is no longer being sold into the public domain. After all, what is going to happen to this software. Is someone going to release it again in 10 years time? No chance. It is dead as far as commercial gain is concerned, so where's the problem? As I say, it is too complex an argument to cover here, but let's have some of your views next time.

Lastly Brad raises the question about New Atari User going quarterly and this is certainly one for discussion but I'll tackle that in the Editorial (hopefully!).

THE INTERNET

Brad Rogers' letter above leads us nicely onto a letter received from Joel Goodwin which wasn't intended for Mailbag but which enclosed an astonishing 'download' from the Internet. As many people have said, there is something for everyone somewhere on the Net and in the comp.sys.atari.8bit section is a Vendors and Developers List of companies and individuals who support the Atari Classic. What is absolutely amazing is that this list contains details of around 138 commercial supporters of the Atari Classic around the world!

Now many of the people listed may have moved on to other pastures but even if half of them are still going that is an astonishing amount of support for a computer that has been out of production for many years. Interesting to note that Atari themselves are not listed!

BULLETIN BOARDS

Alexander Klinner is a new reader from Germany and jumps straight in with some advice regarding Bulletin Boards. "In issue 75 Ray Thompson of Leeds asks if there are any ATARI 8-Bit Bulletin Board Services around. Well, I don't know any in the UK but here in Germany is one called ABBUC-BBS run by the great ATARI BIT BYTER USER CLUB in Herten. It can be called by the phone number +49 (54 39) 32 65 on Monday - Friday from 19.00 to 07.00 and Saturday and Sunday from 15.00 to 08.00 (German time).

The BBS is run by Heiko Bornhorst, who's a very well known person here. Having any problems he surely will help you. The computer used is an ATARI 800XL with 320KB RAM and a hard disk connected by the Black Box. Use BOBTERM or any other software for connecting (you can choose between various protocols) and log in as GUEST. The bad news is that your phone bill will raise to heaven if you call for too long a time and that most of the menus are in German, but many of the sites (= Bretter) are in English, e.g. the news from The Netherlands or the U.S.A.

You can call ABBUC via Internet too on http://wwwcip.rus.uni-stuttgart.de/~infl1492/abbuc/ and you can e-mail anything at 023663 9696-1 @t-online.de

❢ *Many thanks, Alexander, I am sure that we will be hearing more from you in the future.*

HARD DISKS AGAIN

Well, what did I say? The very next letter comes once again from Alexander Klinner in Germany who has more specific information about connecting a hard disk to your Atari. "Paul Kovacs in Darlington asked if it is possible to connect a hard-drive to the 130XE. As you said this is possible, but why buy an expensive Supra interface. Have you ever heard of the Black Box? The Black Box is a T-formed platine which offers

* RS-232 serial modem port
* parallel printer port and
* a SASI/SCSI hard disk port!

That this extension exists and works fine is proved by the ABBUC-BBS mentioned in my earlier letter. Sadly I don't know whether it is available in the UK but you can buy it here in Germany for about DM399 (approx. £177) as shown in the latest catalogue of Ulf Petersen, Postfach 11 03, D-24318 Lütjenburg, Germany. The prices are from 1993/4 but may not have changed much. Originally the Black Box was sold for $199.95 by CSS Computer Software Services, P.O. Box 17660, Rochester, NY 14617, U.S.A.

I would be very delighted if you could spread this information among your readers. Think of all the disks you could save (it is possible to boot from HD - until it crashes down; think of Les!). The Black Box allows hard disks with 20MB to 200MB or more to be connected and offers additional features.

CASSETTE HELP

Much of the same information about hard disk drives also came from Ron Hoffman in Spokane, U.S.A. for which much thanks but Ron is main-



ly concerned about trying to get some Transdisked copies of certain cassettes that he bought a couple of years ago. I am sure that we printed this request quite some time ago but got no response, until now! R. McGrath from Dublin has helped out with Universal Hero, Feud, Bounty Bob and Henry's House but Ron still needs copies of the following: Milk Race, Arkanoid, Storm, Heartache, Twilight World, Thunderfox, Little Devil and Speed Hawk.

Ron has bought all of these cassettes from us and also Transdisk but cannot get Transdisk to work on his U.S. system. If anyone can supply Transdisked copies of these games send them direct to Ron Hoffman at Box 10573, Spokane, WA 99209, U.S.A.

JOYSTICKS

Finally a query from James Martin from Oklahoma "This is a question concerning something that I have been wondering about for some time now. Does anyone know if the Sega joysticks and Joypads are compatible with the Atari Classic?

I am looking for a joypad controller for the 8-bit and the Sega controllers had the nine pin connector just like the Atari, however I haven't tried it. I read somewhere, long ago, that the joysticks were compatible. Maybe someone could do an article on controllers available for the Atari Classic."

Well, has anyone chucked away their Sega and kept the joypad for use on a better computer? Won't take you long to drop us a line if you have.

That's it for Mailbag this issue. Only a few letters but some very interesting points raised and some thoughts for you to write in about. Let's hear from you about the question of commercial software going public domain or anything at all to do with the Atari Classic or the world at large. Allan might be back next time if we can sort out a few logistic problems but in any event, if your letters are in the Mailbag tray for the next issue they will get published. Write! Now!

WRITE TO US!

Air your views on all things Atari or help your fellow users with their queries - even ask for help yourself if you want. It's all interesting, if only you write it down.

MAILBAG
NEW ATARI USER
P.O. BOX 54
STAFFORD
ST16 1TB

DISK BONUS

JURASSIC GOLD

Another great arcade adventure from Bill Halsall

After successfully recovering the treasure in Olde Gold and Indian Gold your fortune awaits you once more in the final part of the trilogy.

Your task is quite straightforward. Rumour has it that there is a lost world, somewhere up river, and it is brimming with treasures you can only dream of. It is believed that there is a diamond the size of a boulder and a correspondingly large gold nugget.

All you have to do is collect both treasures and deposit them where you started. The down side is that there are stories of prehistoric monsters roaming the land. The good news is that nobody has returned to say that they have actually seen any. The bad news is that nobody has actually returned from the river at all!

Use the joystick to move your player and press the button to pick up or drop objects. As before, to kill adversaries you normally walk into them whilst carrying the appropriate weapon. However, on this occasion, things could be easier or harder, but that would be telling. Also, don't throw anything away, you never know when it might come in handy again! If you do happen to get run over by a reptile simply press the button to continue.

If you find yourself in an impossible position and want to start from scratch press RESET then load and run the program again.

Anyway, enough talking. Don your leopard skin and venture forth. At least the odds are better than the lottery - just!

This great program is the BONUS on this issue's disk. If you are not a disk subscriber you can still obtain a copy for £2.95 from NEW ATARI USER, P.O. BOX 54, STAFFORD, ST16 1TB. Please make cheques payable to PAGE 6 PUBLISHING or order by telephone with your Visa or Access card on 01785 241153

PLEASE RELEASE ME

No it's not Dinky-Donk crooning once again but Avram Dumitrescu taking at look at some items of software that never saw the light of day or sparked briefly then faded away

Once upon a time, way back in 1989 in fact, Kirk Ruebottom wrote a fascinating piece on rare and extremely elusive Atari games. Most of the software he talked about could only be bought at expensive import prices from America.

I've taken Kirk's idea further and looked at games that only exist as forgotten sparks in the eyes of old Atari programmers. Why? Is there a point in trying to excite you about concepts and embryos instead of complete programs? I think there is, if only to let you know what may be hiding in a filing cabinet in an unused storeroom, somewhere.

SHOOTING STARS

Around 1990 two new Atari realted companies began their brilliantly bright but abruptly short lives.

Harlequin and ***Fantasy*** both became known in the news pages of New Atari User magazine with snippets of intriguing program names and descriptions seen before only on the more popular eight and sixteen bit machines. Harlequin was headed by Stephen Goss and told the Atari world they were starting to convert a lavish gorge of titles from other computer systems. How does **Paperboy** sound? **Ghosts 'n' Goblins? 1942? Menace? Last Ninja 2?**

Such a prestigious range of software would require the very best machine-code programmers and, after requesting help, Stephen Goss later announced that his plea for an army of logic-loving minds had been very successful. Everything seemed fine up to here - incredible enthusiasm had been induced amongst programmers and the public, negotiations with big software houses had started and the entire Atari community waited for something big. Out of all this came Plastron.

Plastron *had* to be a good game to show how impressive Harlequin could be and it was. Graphically, it looks very pretty and Mike Oldfield would be proud of the music. Wait a minute. Didn't Harlequin claim they were going to *convert* games from other machines? Where is the arcade version of Plastron? What computer did it come from before?

Plastron was an original program and so was their next release, Zero Wars. If Harlequin was to release software in a market where other companies feared to sell then it obviously was not going to sink thousands of pounds into a license for a 'tester' game. Plastron was not that original as the idea was taken from an ancient Atari game called Moon Buggy. While Harlequin did an admirable job dressing up the audio and visuals the gameplay came still wearing velvet flares. Plastron quickly became boring because the simple gameplay does not change with each level.

Nevertheless Plastron was a piece of software Harlequin were proud of. So proud, in fact, they overestimated demand and produced thousands and thousands of copies. And why not? Here was an impressive looking game from a new company with a very shiny future.

Selling software in the 1990s for an eight bit machine is a risky occupation. Trying to make lots of profit from software for a machine that has a community far too small to make millions from is a bad idea.

Plastron did not sell well enough.

Harlequin's next release, Zero Wars, was a disappointment compared to their initial game. It lacked the wild explosion of colour Plastron had and the concept was not realised well enough. It was not fun to play.

After this no more was heard from Harlequin. But let's speculate. Their catalogue of software was two programs with average gameplay. What would we see from them today? Would they become Atari's equivalent of Ocean software? Lots of Ocean eight and sixteen bit software was movie and arcade machine licenses that looked nice but played unimaginatively. Try, if you dare, Chase HQ, Terminator 2 or Total Recall on the Commodore 64.

Conversions are difficult to do as the original movie or arcade machine sets very high standards. Harlequin may have introduced a lot of turkeys but it's almost certain that they, just like Ocean, may have released some super software if only they had survived.

PURE FANTASY?

Fantasy were not quite as successful as Harlequin. Their proposed bag of goodies contained their own brew of games - **Pod 1 and 2, Flux the Flea, Electro Fac** and **10 Downing Street**. Screenshots were forwarded to New Atari User and Les Ellingham commented on how they resembled Zeppelin's best games. Curiosity crept out again but not with the same zeal of Harlequin's birth. It soon crept back to wherever it lurks because Fantasy died without releasing an electronic sausage. Maybe the company name should have told us something?

COUNTING ON MICRO DISCOUNT

Let's move onto something more substantial. Derek Fern has, over the years, released some astonishing stuff. One project was a midi-blaster sound card from Creative Labs. Sound cards give the PC computer the ability to play music and sound effects very, very well (the soundchip ordinary PCs originally have beep a bit but not much else) and so he wired a soundboard to an Atari 130XE. To advertise the hardware Derek offered demo tapes of the

modified Atari playing a selection of songs. Compared to POKEY, Atari's built in chip, the sound is stunning, though some tunes on his tape resemble supermarket muzak.

Shadow of the Beast was another big name game Harlequin intended to release but never did. The programmer, however, has completed a few levels of the game and Derek Fern said his company, Micro Discount, may push for it to be finished and released. The game's selling point was the incredible graphics - the Amiga version had thirteen layers of parallex scrolling and adversaries too large to fit On the screen.

Even more exciting is the Atari's **Commando**. The game plays like Gunlaw - you control a soldier moving vertically up the screen blasting the enemy, throwing grenades and so on. Atari planned to launch Commando on cartridge only. Why? Because one megabyte (1024K - most Atari games are 64K in size) of sound, graphics and code does not fit on floppy disk!

Before you flee to the West Midlands for copies of Commando, I have some bad news. Commando is a prototype cartridge - it has not been finished. The title screen is there, the character you control is well animated but only a few levels exist and any enemies are only fuzzy blocks. I doubt the game will be finished due to its immense size unless, perhaps, it comes out on three floppy disks.

Do you remember Elektraglide? The racing car game with sumptuous graphics by Adam Billyard? Les Ellingham, editor of New Atari User, told me recently programmer Joel Goodwin contacted Adam on the Internet and found out he had some unreleased games. They may Come out soon.

Best Electronics in America bought, from Atari, sixty pallets of hardware and software. A pallet's size varies on its contents but I estimate each one is a three to four foot square, a few feet high. Imagine what secrets are held in Best Electronic's warehouse. Much of the merchandise may be standard Atari hardware and software but it's very likely some pallets contain programs only known inside the deepest and most secret rooms of Atari Corporation.

SOME ESCAPED

Finally, have you ever wondered what programs exist illegally? I have come across lists of games never officially released but which still somehow escaped into the public.

I have heard **Elite** and the **Last Ninja 1** exist. **Indiana Jones** may. **Superpacman** could. **Pastfinder** does (I have played this unreleased game - looks fantastic and is one of the most enjoyable Atari games I have experienced).

Will any of these programs ever be sold? Micro Discount and Page 6 are companies most likely to bring them out. At the moment any new release generates lots of excitement so imagine the frenzy if **Last Ninja 2** and Shadow of the Beast are advertised in the next issue of New Atari User.

As a follow-up to this article I want to investigate Vapourware on the eight bits and VCS systems. Computers that almost hit the shops but didn't quite. Do you know anything about the 65XEP, the 800XE, Atari Cosmos, even the 1450XLD? I also would like to add any more unreleased games to my small list and look at Harlequin, Fantasy and any other interesting Atari company in more detail. If you have any ionteresting snippets of information, write and let me know. My address is:

Avram Dumitrescu
190, Colinmill,
Poleglass, Dunmurry,
Belfast,
Northern Ireland BT17 OAU



PROGRAMMING

HIGH SCORE TABLE

by Andy Guillaume

So you want a Disk-based High Score Table for your Turbo Basic programs? Here it is, dispelling the myths surrounding 'complicated' routines in one stroke.

All of the scores are held in the array S(n) where n is the position in the table, the names being held in NM$. After a game score, SC, has been found and a name input the program just works from the bottom of the table upwards comparing SC to S(n) for each entry. If SC is greater or equal to the current S(n) score then this score is moved down one entry and the current S(n) set to SC. The name is also moved and updated in the NM$ array. There has to be one more entry in the table than required so that the last entry has somewhere to be moved to, thus the S array is dimensioned to 11 elements and NM$ to 110 elements (10 characters per name).

The Load routine retrieves a previously saved table from disk. Take out the REM on line 80 after a table has been Saved to make the program Load the table on each run. The Save routine does just what it says, storing the table on disk. Take out the REM on line 280 so that the table is saved after each new entry.

Note that the PLAYERS SCORE AND NAME section of the program should be replaced with ... your game!!

```
TO 1 REM ###########################
VN 2 REM #      HIGH SCORES        #
XG 3 REM #     (TURBO BASIC)       #
OO 4 REM #   by Andy Guillaume     #
KA 5 REM # ----------------------- #
EG 6 REM # NEW ATARI USER - JUL 96 #
TU 7 REM ###########################
NN 8 REM
JY 10 REM  SETUP TABLE
JS 20 DIM N$(110),S(11),NM$(10)
KX 30 FOR N=%1 TO 11
WQ 40 P=(N*10)-9
OL 50 N$(P,P+9)=".........."
MU 60 S(N)=1100-(N*100)
KO 70 NEXT N
GK 80 REM EXEC LTABLE:REM  << LOAD TABLE
LL 90 REM  SHOW TABLE
OA 100 GRAPHICS %0
WG 110 FOR N=%1 TO 10
YG 120 P=(N*10)-9
RS 130 POSITION 7,5+N:? N
LR 140 POSITION 10,5+N:? N$(P,P+9)
KQ 150 POSITION 22,5+N:? S(N)
IB 160 NEXT N
LI 170 REM  PLAYERS SCORE AND NAME
UP 180 POSITION %0,18:INPUT "ENTER SCORE"
   ;SC
AV 190 POSITION %0,19:INPUT "ENTER NAME";
   NM$
SI 200 REM  UPDATE TABLE
WR 210 FOR N=10 TO %1 STEP -%1
YH 220 P=(N*10)-9
WS 230 IF SC<=S(N) THEN 270
OW 240 S(N+%1)=S(N):S(N)=SC
DL 250 N$(P+10,P+19)=N$(P,P+9)
IB 260 N$(P,P+9)="          ":N$(P,P+9)=N
   M$
IE 270 NEXT N
WS 280 REM EXEC STABLE:REM  << SAVE TABL
E
LZ 290 GOTO 100
```

continued on page 23



SUMMER GAMES

It seems a long time since we had some major reviews of recently released software so this batch of reviews from Daniel Baverstock is most welcome

There's nothing like trying to achieve the all over sun tan. The hour or two (whose counting?) seems to creep by. Every few minutes you glance to see your skin turn even whiter than when you first donned your shorts! It's only when you accept the fact that you never tanned anyway, that you discover the few apparently ineffective hours of sun light has turned you into a human Flamingo!

As ever you pledge you'll never bask under the great sun lamp in the sky again. You undergo the gradual and sometimes painful reptilian shed of skin, making you an ideal candidate for a B-movie monster! Before your ordeal is over you find yourself avoiding any social interaction until your skin resumes it's familiar 'Persil white' complexion. In the end you have to ask yourself is it worth all the time and lotion, (sorry, I mean commotion!)?

Well, I am sitting in the sun on this Wednesday afternoon, writing this account of what I shall be thinking of in a few hours time. Time, I think, for a few reviews of some new and not so new 8-bit games.

For the last few months I have been rather inactive in the Atari arena, so the number of titles to review has steadily been rising, giving me a nice selection to babble on about. I recently bought a few titles off a friend and after an extremely long wait, (thanks to Royal Mail's usual incompetence I suspect), some new titles from Micro Discount. I have ***BARBARIAN, STARBALL, CRYPTS OF EGYPT*** and ***TUBE BADDIES*** in my sights, covering a wide range of different gameplay styles.

BARBARIAN

The personally eagerly awaited Barbarian was the first to be viewed. Undoubtedly Barbarian is derived from the Barbarian I and II games that appeared on most 8-bit formats, (as well as the Amiga), in the late 80's.

Barbarian is a single screen one to two player hack 'em 'up, pitting two sword bearing Arnold 'Conan' Schwarzenegger look alikes against one another in an arena. The game views the combat from the side in a 2-D perspective like most other beat 'em 'ups, Mortal Kombat for example.

You can choose between one or two players on the title screen before playing. The title screen itself has the Barbarian logo, and the credits below, with snakes twisted around swords at either side. A suitably appropriate and fairly catchy chip music plays.

The game loads in a few seconds, randomly loading one of three possible arenas, giving the backdrop locations a little variety. The arena uses up under half the lower screen, with the title, players' energy levels and a sixty second time limit displayed above.

I was very surprised to find the whole screen, (title, arena backdrop and characters), were all in one colour scheme. This is a mistake since the characters, though distinguishable from the background, don't stand out like they could. The programmers haven't done the game any justice since the colours could have been far more exciting.

Sound effects are even more disappointing, almost non-existent and those used less than adequate for the task of the likes of 'clashing' swords, ('tinkle' is a better word to describe it!).

The game is initially played better in two player mode. Both players then have a chance of getting accustomed to the moves at their disposal which include various overhead, mid, and lower sword hacks, two blocks, left and right ground rolls, a shin kick, crouch, jump, and a great finishing move. Your Barbarian pivots like a ballerina on his feet, wielding his sword over head before beheading his opponent, whose corpse fall to the floor as his head spins off the screen!

Playing against your inactive second player gives you even more practice, and while happily hacking away at him, I found out a further disappointment in the gameplay. The game duration is way too limited! Once you kill your opponent in the first round, (remember sixty seconds a round), you have another two rounds before the game finishes. You are returned to the credits to enter your name. That's it! Only three minutes of play!

The game has some close similarities with International Karate, (aka World Karate Championships), which at the same time overshadows Barbarian completely. For ex-

ample, the characters, if left, face and talk to you like those in International Karate, and when they win, they throw their sword arm up in victory.

Directly in comparison with International Karate, Barbarian's limited three round gameplay is the biggest mistake and ultimately the Achilles heal of the game. Yet it never had to be! International Karate uses the disk space to load many screens of colourful backdrop locations, while you progress through the increasingly difficult levels. This incentive of progression and challenge is totally lacking in Barbarian.

Other than to beat the computer opponent (almost impossible), beat player two or simply to better your score (entered after the three rounds but not saved) there is hardly any reason to play the game. Because your computer opponent is so hard to attack, the moves at your disposal are almost never used. I found myself continually being kicked to the ground in my screen corner. It completely eliminated any feeling of skill or combat as I wobbled the joystick in circles to no avail!

All games of this kind rely on this fundamental element of gameplay, for without it there is no game. I suspect the abrupt ending of gameplay is because the game was most likely made for primarily the cassette market and a 64K memory limitation. If this is the case, the quality of gameplay should have taken precedence over making a game for tape limitations only. If however, this was targeted for disk drives, this is a needless waste of the 128K available on a disk side! What's the point in owning a drive with increased speed and storage specifications over that of dated cassette storage if the games never use it?

Had the game been fully optimised, it would have utilised the whole disk side, (and perhaps side two as well), would have loaded a variety of colourful backdrops, had difficulty settings, and sampled sounds. International Karate pretty much managed it, so what's the problem? I put that question to future software developers.

If their counter argument is on the basis of financial investment and incentive, I would argue that had this game satisfied all the criteria I have mentioned, the resulting gameplay and challenge would prove more popular and give a better financial return than making a uninspiring and limited game more accessible to cassette based users.

As for a 64K memory limitation, if the great platform adventure Miecze Valdgira II can fit it's enormous level size and countless characters (each with fantastically detailed animation) into 64K, then surely a detailed one screen beat 'em 'up is more than possible?

I always think that new releases should be an improvement on old software. Regrettably it is not the case with Barbarian.

A welcome attempt at a long needed beat 'em 'up, but despite the great £4 price tag, the good animation and moves, it's a big let down. Recommended for two players only.

STARBALL

Starball is an aerial perspective 360 degree scrolling arcade puzzler. You control Starball, who is deflating constantly as it's air escapes, acting near enough as your level timer to keep you on your toes. The collection of crystals scattered around the place allows you to constantly replenish your air for as long as they last, while you set about the task of negotiating gaps in the floor through which you fall to the ground far below, and grabbing hands that periodically emerge from the ground to pull you under.

The aim is to collect all the crystals scattered around and locked away by obtaining green and red keys and a few other items located inside the locked chambers. Control of Starball initially couldn't be any simpler. Push in a direction and it moves. Soon, you find out it isn't that simple, as your speed and inertia increases and everything you hit while moving causes you to rebound in the opposite direction.

The precise joystick control you have to have over Starball can prove frustrating. You sometimes teeter precariously over the edge of huge gaps in the ground, and may have to move slowly despite your decreasing time limit.

The various chambers containing keys have to be opened in the correct sequence. You may open a chamber, and therefore waste keys, to find you can't collect enough in that chamber to open up any of the remaining ones due to the number and combination of green and red doors locking any one chamber. The starting level's first key is found in a chamber with a constantly opening and closing door which has to be quickly negotiated or Starball is squashed.

Gameplay is viewed in the top two thirds of the screen, the bottom being the status panel where score, air level, life status, and the number of red and green keys you have is shown.

You have three lives available, the level loading each time you run out of air, fall through a gap in the floor, or get pulled under by a grabbing hand. Each time you die you start right back at the beginning, adding to the frustration.

Not having passed the first level, I am only currently aware of two items. A dynamite plunger used to blow a space three by three square opening up a hole in another chamber, and a hammer, the purpose of which I have yet to discover. Both are found in chambers and are activated by rolling over them and in the case of the plunger then getting out the way, fast!

The chip music from the title screen and gameplay is fairly standard but catchy nevertheless. Unfortunately sound effects are few and none too inspiring. The graphics aren't bad at all. From the blue starfield title screen to the in-game visuals, everything is pleasing to the eye.

The Starball itself is small and shaded grey. Your level is suspended way above the ground, indicated effectively as the scroll for the ground and platform move at different speeds, and you can see the ground below through the gaps. The chambers are patterned blocks, while the floor is stripey. Icons and crystals are drawn neatly and the animation of the hands is quite a nice touch as they reach to grab and pull you under.

Overall, this game will appeal to those who have a talent for precise joystick control and are quick thinkers. The gameplay is challenging and rewarding, if frustrating at times. As I can't even pass the first level, I must assume there are many levels to keep you busy.

I shall no doubt come back to this game now and again, for it's one of those games that I enjoy in small doses. Two much of it when you keep starting at the beginning after falling through a gap in the floor for the tenth time may cause violent temper tantrums! Not speaking from personal experience of course, being a gentle and patient man!

CRYPTS OF EGYPT

Another pixel perfect game, this time a platformer, requiring even more precise control over yet another tiny sprite! You control a smaller version of Zeppelin's Cavernia character. You know, the guy with the backpack and hat with the life of it's own!

The title screen is nicely put together, rather

like the small scrolling mini demos from European demo crews. The in-game graphics are very good, almost identical to the style of Cavernia only with Egyptian undertones, using lots of shades of yellow.

I made sure I had sufficiently played the game to the limits of my playing ability before reviewing it but actually I have only got to the fourth screen! Like the environment in Deimos, the Crypt is very large and detailed, with many traps which I continually set off by tripping on the switch! Moving spikes and flickering fire pits have to be contended with. Most have a switch that can be turned on or off by throwing a pebble at them, which you have plenty of in your pockets. Some traps can only be turned off temporarily, sometimes from another screen, so precision and speed are essential.

A large green ghost, (sounds more and more like Cavernia), haunts parts of the screen, and can be temporarily vanquished with your trusty pebbles, though he'll reappear a few seconds later somewhere else. After exiting the screen, the next part of the Crypt is ready for you contend with, sometimes with exits in all four directions. Exploration of all available chambers is essential. It can lead to switches that deactivate traps in the previous screen, as mentioned earlier.

Music and sound effects are adequate, but they always are with games of this kind. Pity! Again, the frustration factor is present, even more so since you only have one life. You have an energy bar which depletes when injured although small vase like artefacts can be collected to replenish your energy levels.

I liked this game from the moment of loading. Greatly detailed and colourful graphics, well animated sprites, and extremely frustrating gameplay requiring precise platforming control! I found it strangely compelling to play. Crypts of Egypt joins Deimos and Cavernia among the ranks of my favourite frustrating platform games.

TUBE BADDIES

Tube Baddies is a true arcade game. It is an old Atari UK game that never got the release it deserved. Programmed by well known programming geniuses Ian Macintosh and Richard Munns, Richard Gore and Micro Discount obtained the rights to publish it last year.

It has such simple yet addictive gameplay and colourful graphics which, combined with some excellent chip music, makes for a greatly playable arcade game.

The basic story introduces you to Bip and Bop, the handymen of the Tubular underworld (two characters that look like small blobs of gum). They have the job of fixing all the pipes, which have been falling apart from neglect and from the unwelcome attention of the Tube Baddies, small metal munching creatures that make the holes on the pipework. The typical 1980's Atari arcade puzzler plot and characters really! Armed with a stungun and a bucket called Barney you have to rid the levels of these creatures and fix the pipes.

The title screen and music is great, colourful and frenzied. Here you can select two players, although the game is as equally playable with one player. The levels themselves are very colourful and detailed. The pipework twists around the screen and Tube Baddies emerge from existing holes to patrol around like menacing purple road cones wearing shades making holes wherever they go.

Below, Barney the bucket, who is cutely drawn and well animated rolling his eyes and displaying various facial expressions, awaits any Tube Baddies who you stun, to be deflected into the bucket. A counter indicates the number successfully 'binned' in the bucket, while those missing Barney go back into the pipework.

The more Tube Baddies collected the more bucket bonus points you'll collect every two levels. The bucket bonus is timed, so if you don't complete the game before the bonus timer runs out, it gets set to zero, and you lose all the bonuses collected. Patching up pipework is accomplished by pressing fire Çover a hole to deposit a cartoon style plaster over it, a nice touch.

After stunning a Tube Baddy, they'll continually rebound off the sides until you can fire them into the bucket. A level is completed when you patch up the pipes and eradicate all the tube baddies from it. Patching of pipes isn't essential for level completion, but worth it while capturing existing tube baddies as it stops any additional Tube Baddies from emerging into the level.

There are various hazards to contend with. Chewing gum and spider's webs slow you down as you pass over them. Walls restrict you, but not the Tube Baddies, so you have to be careful not to get trapped, especially if they are stunned, as they move faster than you. Later on, electric cables which emit bolts of electricity make things more difficult. Careful timing is required here.

Other items can be picked up, some detrimental and some beneficial to your health. A joystick icon reverses your controls for a few seconds which can be very bewildering, while a backwards 'S' icon skips you to the next level. You have four lives and any contact with Tube Baddies, the hazardous environment or a dangerous item causes loss of life.

The overall quality of the game is very high, from the rainbow-esque colours and jingle style music, to the increasingly challenging variety of levels.

The bucket bonus screens are amusing, where the Tube Baddies are hotly pursued by Bip and Bop in their cars over the top of the screen. The graphics are really chunky, but so colourful that it doesn't matter, as are the Hi-score input and display screens.

The whole game has been programmed with all gaming aspects cleverly considered. Playing the game is great on your own, and even better when you are working both in competition for points and in co-operation to complete the level. There are plenty of challenging levels to keep you engaged in this game for a long while.

In my mind Tube Baddies is the best of this particular bunch and comes highly recommended!

All the games reviewed are available for £4 each from Micro Discount. Barbarian, Starball and Crypts of Egypt are copyright of Poland's MIRAGE software, while Tube Baddies is copyright of Atari U.K.

No, onto more pressing matters, my tan! I don't seem to have even been touched by the sun. Not one shade of crimson? Perhaps it has something to do with the fact that it has been raining for the past hour?

HIGH SCORE TABLE

continued from page 17

```
EF 300 REM  SAVE TABLE
CR 310 PROC STABLE
US 320 OPEN #%1,8,%0,"D:SCORES.TAB"
WM 330 FOR N=%1 TO 10
DP 340 %PUT #1,S(N)
IB 350 NEXT N
JT 360 BPUT #%1,ADR(N$),100
EM 370 CLOSE #%1
WB 380 ENDPROC
XY 390 REM  LOAD TABLE
ZY 400 PROC LTABLE
SN 410 OPEN #%1,4,%0,"D:SCORES.TAB"
WL 420 FOR N=%1 TO 10
LD 430 %GET #1,P:S(N)=P
IA 440 NEXT N
DI 450 BGET #%1,ADR(N$),100
EL 460 CLOSE #%1
WA 470 ENDPROC
```

HEY! HEY! it's The TIPSTER

Last issue John Hull asked for some info on Mercenary and, lo and behold, it all appeared. Along with some clues Roald Sund from Norway sent in some fabulous maps, probably the best maps we have ever had, so we are holding over the tips promised last time to next time and are devoting this issue's Tipster to Mercenary - maps, clues and info.

MERCENARY 3?

Graeme Fenwick up north in Dundee hopes he can answer the question about the existence of Mercenary 3 with this information. He says: In response to the MERCENARY question last issue I'm slightly confused as to which game John Hull is talking about when he refers to Mercenary 2. Does he mean Damocles or, as I suspect, The Second City? Anyway to help clarify things, here's a list of all the Mercenary related stuff available as far as I know.

MERCENARY (1) - the original, available on the Atari 8-bit amongst others

THE SECOND CITY - add on dataset for Mercenary 1. The Second City has never, to my knowledge, been officially referred to as 'Mercenary 2'. Available on Atari 8-bit and others

DAMOCLES (MERCENARY 2) - I've never played this, but as far as I know it's 16-bit only (ST and Amiga at least). Billed as 'Mercenary 2' it featured solid objects and a whole solar system

DATA DISKS FOR MERCENARY 2 - don't know much about these but, if I remember correctly, there were two of them. Presumably same formats as Mercenary 2

MERCENARY III - The final game available on ST and Amiga. Not such a radical difference this time (as far as I know) but I've got this for my Amiga and it was great fun! No datasets available as far as I know

So there you have it. As far as I am aware only Mercenary and The Second City are available for the Atari 8-bit.

MERCENARY

MERCENARY

MERCENARY

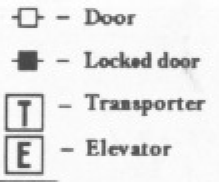
This issue's major contribution comes from Roald Sund in Norway who supplied the maps and this info: I have read in New Atari User, issue 76, that John Hull wants some info about Mercenary, games 1 and 2. I am not sure, but I suspect that what he refers to as Mercenary 2, really is "The Second City", which actually is not Mercenary 2, but more like an addition to Mercenary 1. There is a Mercenary 2 game, but that is known by the name Damocles. I have both Damocles (Mercenary 2) and Mercenary III on disks for the Atari ST. I doubt that any of these games exist for the Atari 8-bit (XL/XE).

I have completed Mercenary - The Second City a long time ago and I also made some maps. I have not found a use for all of the objects, and the maps are not 100% complete, but I am sending them in the hope that some clues can be found there.

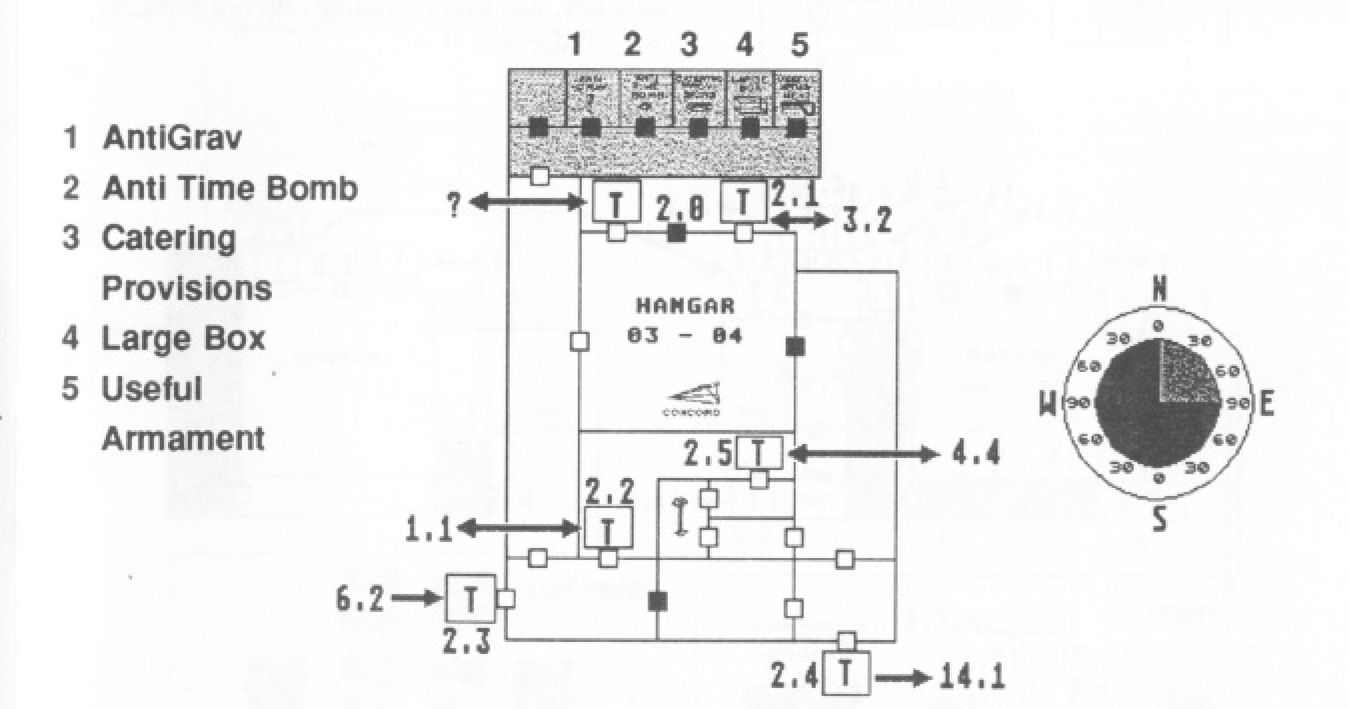
MERCENARY - THE SECOND CITY
Map of "The Underground Complexes"

1.1 ? 2.2 1.0 15.0 PRISON HANGAR 08 - 01 4.5 1.2 PALYAR BRIEFING ROOM BANK 6.1 1.3

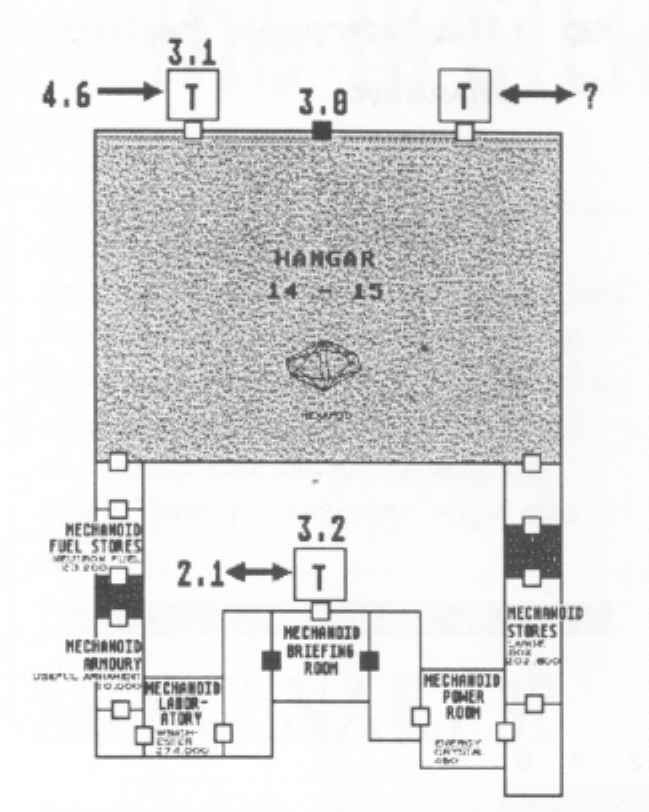
EXPLANATION:





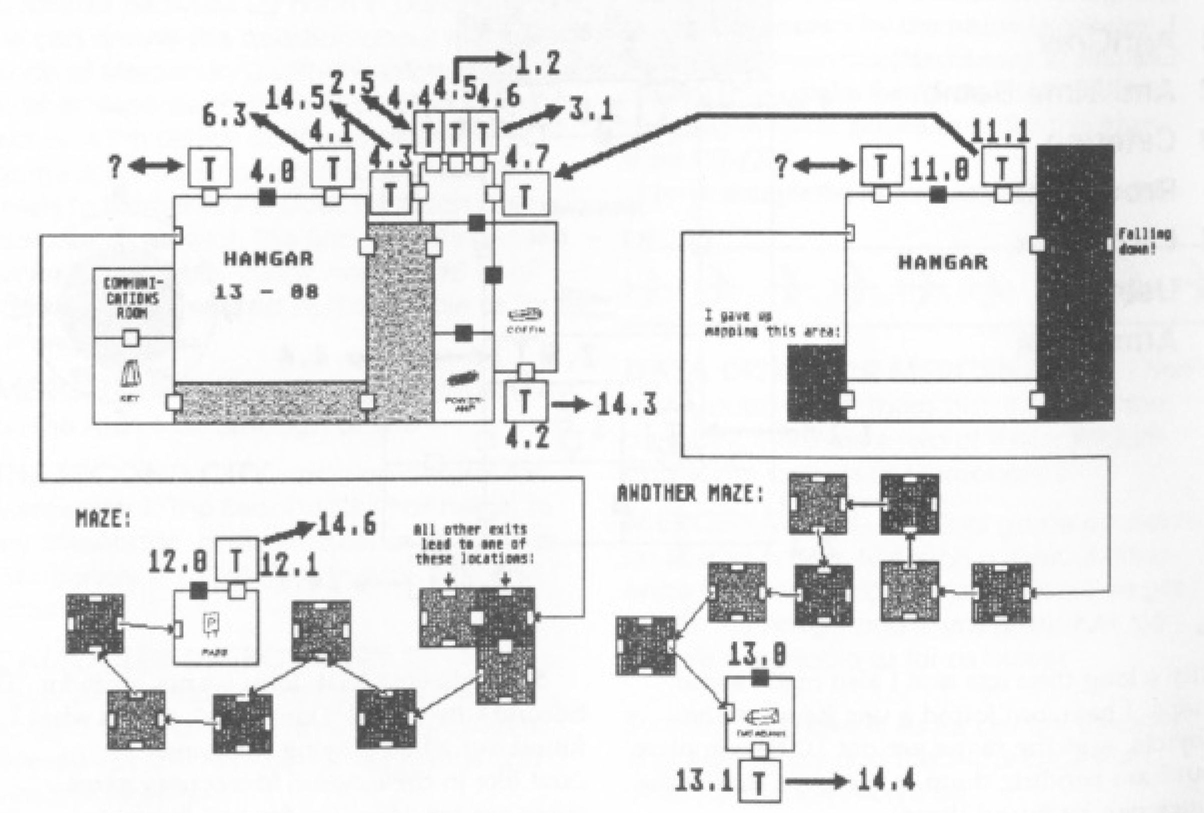


Still believing that John wants hints for "The Second City", NOT "Damocles", this is what I found out when playing the game:
Just like in the original Mercenary game, there are two ways (as far as I know) to



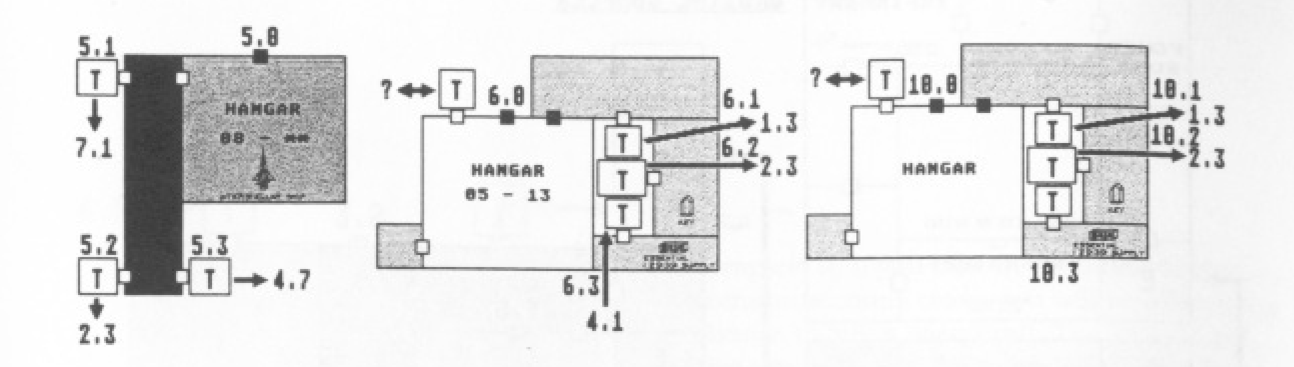
complete it. If you take the antenna to the communications room, you will be offered the chance to buy a spacecraft. The less expensive way is to find the interstellar ship. You need to carry the novadrive to be able to take off with the ship, and you should also have the pass to be able to bring the ship to the surface. (Or you could carry the ship to another hangar where the elevator does not need the pass to work. To carry the ship you need the antigrav, and you will also need the right key to get through all the doors.)

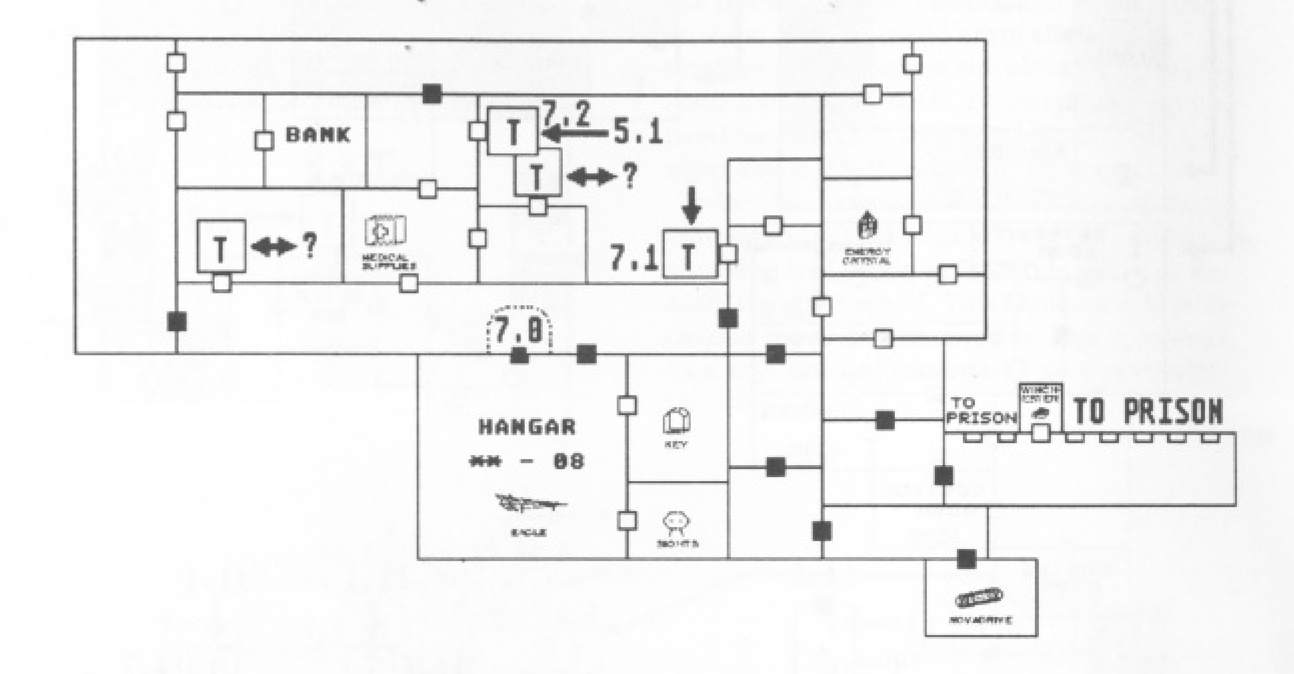
To be able to walk through all locked doors, you should pick up one of your own bullets by shooting and pressing "T" while flying at exactly the right speed. This bullet needs to be carried together with another object to work as a key, but unfortunately I do not remember what object it is.



ORBITAL COMPLEX

INFIRMARY
POWER ROOM
(Room with no exits)
LABORATORY
E
HANGAR
8.0
STORES
ARMOURY
E
INTERVIEW ROOM
BANK
EXCHEQUER
CONFERENCE ROOM
KITCHEN
CONTROL ROOM
ENGINE ROOM
E
9.0
STORES
ARMOURY
THE PALYAR COMMANDER'S BROTHER-IN-LAW'S ROOM
14.0
Falling down!
14.1
2.4
12.1
11.1
14.6
14.7
14.5
14.2
4.3
14.4
14.3
13.1
4.2
7.2
(7.2)
All other doors with this shape lead here!
1.0
2.0 . . .
14.0





NEXT TIME

As indicated the tips promised last time have been squeezed out but look out next issue for tips on **INTERNATIONAL KARATE**, **MIRAX FORCE**, **SPELLBOUND**, **SPEED HAWK**, **YOGI BEAR AND THE GREED MONSTER**, **CAVERNIA**, **OLYMPIC SKIER**, **STAR RAIDERS II**, **DAVID'S MIDNIGHT MAGIC** and **PROTECTOR *PLUS*** any that YOU send in in time. These tips come courtesy of ***Miguel Letemplier***, ***Jason Kendall***, ***Daniel Yelland***, ***Graeme Fenwick*** and ***C. Ayres*** but don't let them grab all the glory, send in some of your own.

KEEP 'EM COMING

Roald Sund's maps were fabulous this time and we have a few tips in stock for next time but don't sit back and get complacent. The Tipster is a hungry little puppy and needs more tips each time, so do the usual and send him something. Pop it in the post to

THE TIPSTER
PAGE 6 PUBLISHING
P.O. BOX 54
STAFFORD
ST16 1DR



ROCKMAN

reviewed by Paul Rixon

The Mirage Software label usually indicates a graphically advanced game and Rockman - a 1995 release from the Polish company - is no exception to the rule. The boot sequence is accompanied by an incredibly large digital read-out representing the count-down to program launch. This pleasant visual touch sets the scene for a game that's characterised by brilliant graphics and music. To demonstrate the point, the game includes not one soundtrack, but TWO! There's an option to choose your personal preference before the action commences.

Visual and sonic feats aside, Rockman is a fairly ordinary game. It's based on a 'grab the goodies and navigate the nasties' concept with the necessary ingredients of platforms, doors, keys, bombs, ammo, lives, bonus points and intriguing objects of the good, bad and sometimes indifferent variety. No doubt there's also a complex story to justify the effort, but unfortunately I wasn't supplied with an English translation!

Each level comprises a single screen design, presenting a unique layout of platforms and objects. Rockman, the character, takes the form of a humorous green bouncing face. By issuing joystick commands, your task is to assist Rockman as he leaps onto ledges, skims along platforms and hops heroically over various unpleasant features. Timing is everything, as they say, since contact with anything nasty swiftly results in the short but accurate message - "You're Dead - Try Again!".

Despite the irritation factor of ever-present lethal objects, the smooth movement of Rockman contributes towards a generally user-friendly feel. The nasties can be overcome by careful avoidance tactics or zapped with a blast from your character's stun-gun. The ammunition is limited though, so it's not wise to be trigger-happy. Once you've collected the goodies, an exit point to the next level is revealed. Status lines at the foot of the screen record your remaining lives, ammunition and bonus time allowance, together with the current score and game level.

To recap, Rockman is a desirable game. It's reasonably addictive, is hard to criticise on playability, and includes the welcome feature of a highscore table. These positive attributes are further enhanced by the high quality of graphical and musical support. Incidentally, I'd like to see what the obviously talented authors can do with a game involving more substance. No doubt the answer lurks among the wide range of imports supplied by Micro Discount.

Title:	**ROCKMAN**
Publisher:	**Mirage**
Supplier:	**Micro Discount**
Format:	**Disk**
Price:	**£4.00**



PROGRAMMING

CHARACTER SET COPIER AND REDEFINER

A redefined character set in the blink of an eye? John Foskett shows how it can be done

There have been many machine code routines published over the years which copy the character set into RAM, but they all leave you to redefine the chosen characters yourself, normally by loading DATA via a loop. Once the character set has been copied into RAM, you first have to find the starting address of the group of characters that you wish to change which is done by finding the INTERNAL code of the first character of the group. The internal code is then multiplied by 8 because each character is made up of 8 bytes and this is then added to the new RAMTOP value which itself must first be multiplied by 256. Apart from the necessary arithmetic which itself can be very confusing, the main problem is that the more characters you wish to redefine, the longer your program takes to initialise and so you may end up with a compromise by redefining just a few characters and making do.

THERE IS AN ANSWER!

The machine code routine presented here first copies the character set into RAM after you have first lowered RAMTOP by at least four pages and then it will redefine your chosen characters up to a maximum of 32 from data stored in the form of a string, instantly, at machine code speed. This routine does not require any arithmetic either, it just requires the internal code of the first character of the group you wish to redefine and the routine will do the rest. You don't even need to point CHBAS (location 756) to your new redefined character set and you don't even need to know the value of RAMTOP itself. For convenience, the routine returns the value of RAMTOP to BASIC via the USR commands equating variable.

```
HY 10 REM ******************************
XH 20 REM *  CHARACTER SET COPIER AND   *
DK 30 REM *          REDEFINER          *
IN 40 REM *   A MACHINE CODE ROUTINE    *
CU 50 REM * WITH DEMONSTRATION PROGRAM  *
CB 60 REM *  BY JOHN FOSKETT   JAN'96   *
CW 70 REM *-----------------------------*
RH 80 REM *    FOR NEW ATARI USER       *
IG 90 REM ******************************
OI 100 DIM CHSET$(88)
VY 110 CHSET$=")[,](<E>K<E>M<E>U"[D]
   -t[B]<E>L%j<E>N1K<Q>MHPyfLfNJPrhhh
   <E>K<E>L[F]K[F]K[F]K"[E]FLJP<;>
   %L[X]ej<E>Lh<E>Nh<E>Mhh<E>01M<Q>
   KHDOPw%j<E>T<M>t[B][.]"
QQ 120 REM
QS 130 REM
FU 140 REM  DEMONSTRATION PROGRAM
UX 150 RT=PEEK(106):POKE 106,RT-4
HW 160 GRAPHICS 0:POKE 16,64:POKE 53774,6
   4:POKE 752,1:POKE 82,2
YS 170 REM  Set-Up CHAR$. 256 Bytes Max
SI 180 DIM CHAR$(208)
QI 190 CHAR$="[ESC,UP]60x00[ESC,BACK SPACE]
   [,][L][L][X][,][,][,][,][,]
   LLf[,][,][,][,][,]33f[,][,][,]
   [,][,][,][ESC,INSERT][,][,][,][,]
   [,][,][,][ESC,INSERT][,]
   [ESC,INSERT][,][,][,][,][A][C][G][G][G]
   [G][C][A]<,>@<.><.><.><.>@<,>
   [C][O]?[ESC,INSERT][ESC,INSERT]?[O]
   [C]3fL[ESC,INSERT][ESC,INSERT]Lf3Lf
   3[ESC,INSERT][ESC,INSERT]3fL@p!
   [ESC,INSERT][ESC,INSERT]!p@"
OP 200 CHAR$(97)="[X]<[ESC,BACK SPACE]
   [ESC,INSERT][ESC,INSERT][ESC,BACK SPACE]
   <[X][ESC,INSERT][ESC,INSERT]CCCC
   [ESC,INSERT][ESC,INSERT][,][,][,][G][O]
   [ESC,UP][X][X][,][,][,]<.>p8[X]
   [X][X][X][ESC,UP][O][G][,][,][,]
   [X][X]8p<.>[,][,][,][,][,][,]
   [,][,][,][C][C][,][,][,][,]
   [,][,]@@[C][C][,][,][,][,][,]
   [,]@@[,][,][,][,][,][,]CC<<<<C
   C<Y>ff<Y><Y>ff<Y>"
OP 210 CHAR$(193)="L<Y>3fL<Y>3f3<Y>Lf3
   <Y>Lf"
XH 220 REM  Call M/C Routine. 97 is the
RE 230 REM  Internal Code for lowercase
QW 240 REM  letter "a"
KT 250 RAMTOP=USR(ADR(CHSET$),97,ADR(CHAR
   $),LEN(CHAR$))
LT 260 REM  Print Screen
KF 270 POSITION 3,0:? "cCHARACTER SET COP
   IER AND REDEFINERd   ffffffffffffffffff
   ffffffffffffffffff"
ZI 280 ? "[ESC,TAB] cWRITTEN BY  JOHN FOS
   KETTd":? "[ESC,TAB]  eeeeeeeeeeeeeeeee
   eeeeeee"
TY 290 ? "  IN THIS DEMONSTRATION PROGRAM
   , ALL   THE LOWERCASE CHARACTERS HAVE
    BEEN   REDEFINED AS FOLLOWS...."
VU 300 ? :? "  a b c d e f g h i j k l m
   n o p q r":? :? "  s t u v w x y z"
GK 310 FOR I=2 TO 10:POSITION 3,I:? "| ":N
   EXT I
OI 320 ? " q[R][R][R][R][R][R][R][R]
   [R][R][R][R][R][R][R]r":POSI
   TION 19,10:? "o[R]iSOME EXAMPLESI[R]
   [R][R]"
LM 330 POKE 82,3:POSITION 3,12:? "o[R][R]
   [R][R][R]p":FOR I=0 TO 7:? "|wwww
   w| ":NEXT I:? "q[R][R][R][R][R]r"
RZ 340 POKE 82,10:POSITION 10,12:? "ca100
   00d":? " ffffff"
NZ 350 POSITION 10,14:? "s[N][N][N][N]
   [N][N]t":FOR I=0 TO 2:? "[B]yyyzzz
   [V]":NEXT I
MG 360 FOR I=0 TO 2:? "[B]zzzyyy[V]":NE
   XT I:? "u[M][M][M][M][M][M]v"
```

continued

Underline = INVERSE CHARACTERS · [ ] = CONTROL + CHARACTER · < > = INVERSE CONTROL + CHARACTER

USING THE ROUTINE

The routine is used via the USR command and is configured as follows....

RAMTOP=USR(ADR(CHSET$),INT,ADR(CHAR$),LEN(CHAR$))

Where....
CHSET$ is the machine code routine.
CHAR$ is the redefined character data as a string.
INT is the internal code of the first character to be redefined

```
LX 370 POKE 82,19:POSITION 19,12:FOR I=0
   TO 3:? "n[R]n[R]n[R]n":? "| n| n| n| ":
   NEXT I:? "n[R]n[R]n[R]n"
EV 380 POKE 82,27:POSITION 27,12:? "i[R]
   [R]jj  kk[R][R]l"
DZ 390 POSITION 27,13:? "s[N][N][N][N]
   [N][N][N][N][N][N]t":FOR I=0 TO
    3:? "[B]xxxxxxxxxx[V]":NEXT I:? "u[M]
   [M][M][M][M][M][M][M][M][M]
   v"
OZ 400 FOR I=0 TO 2:? "jkjkjkjkjk":NEXT
    I
YU 410 POKE 82,0:FOR I=0 TO 20 STEP 2:POS
   ITION 0,I:? "zy[V]":? "yz[V]";:NEXT
   I:POSITION 0,22:? "zyzyzyz":? "yzyzyzy
   ";
AR 420 POSITION 7,22:? "gSTART hRUN AGAIN
   ":POSITION 7,23:? "gSELECThSEE LOWERCA
   SE CHARACTERS";
FX 430 I=PEEK(53279):IF I=5 THEN POKE 756
   ,224:FOR I=0 TO 400:NEXT I:POKE 756,RA
   MTOP:GOTO 430
BF 440 ON I<>6 GOTO 430:POKE 106,RT:RUN
```

Underline = INVERSE CHARACTERS · [ ] = CONTROL + CHARACTER · < > = INVERSE CONTROL + CHARACTER

RAMTOP is the variable returning the new RAMTOP value to BASIC.

When using the routine within your own programs, RAMTOP must first be lowered after which a GRAPHICS call is actioned in the normal way as follows....

```
100 I=PEEK(106):POKE 106,I-4
110 GRAPHICS 0
120 DIM CHSET$(X),CHAR$(Y)
130 CHSET$="....
140 CHAR$=".....
150 RAMTOP=USR(ADR(CHSET$),INT,ADR-
(CHAR$),LEN(CHAR$))
```

HOW THE ROUTINE WORKS

The source code can be divided in two sections, both of which are described separately as follows.

CHARACTER SET COPIER

This section is responsible for copying the character set from its position in ROM into a position in RAM occupying 4 pages in RAM above the previously lowered RAMTOP. It is important to note that, in effect, RAMTOP (loaded from location 106) and CHBAS (loaded from location 756) are the high byte values when considering them with respect to the normal 2 byte addressing. Although the low bytes do not actually exist because both RAMTOP and CHBAS must begin on a page boundary, they must be considered as zero. CHBAS and RAMTOP are loaded into the zero page locations 204 and 206 respectively and the corresponding low byte addresses are 203 and 205 which are loaded with zero. Because both now have high and low bytes, it can be seen that it is a simple matter to copy from one to the other via a loop.

The loop (LOOP1) copies one page at a time (256 bytes) by indirect indexed addressing using the Y register after which the high bytes are incremented so that the loop can be repeated to copy the next page. After copying all 4 pages that is when the X register has been decremented to zero, this section of the routine is exited and the next section actioned.

CHARACTER REDEFINER

This section is responsible for redefining the chosen characters and reuses the same page zero addresses as used before by the previous section, loading them with the three parameters used in the USR call. Location 203 and 204 is used to store the internal code of the first character of the group to be redefined, but note that both locations at this stage contain the same value. Location 205 and 206 are used to store the address of the redefining string (CHAR$) in the normal 2 byte lo/hi format and location 207 is used to store the length of the string.

Because the characters are made up using 8 bytes each, the internal code loaded into location 203 and 204 must be multiplied by 8. The multiplication is achieved by shifting the contents of location 203 three positions to the left using "ASL 203" and the contents of location 204 to the right five positions using "LSR 204" effectively dividing the contents of this location by 32. Since location 204 is the high byte, its contents are effectively multiplied by 256 and dividing its contents by 32 in effect multiplies by 8 (256/32=8). Shifting the contents of both locations in this way multiplies by 8 as if the contents of location 203 was shifted within a 16 bit byte using location 204 as the corresponding high byte. The next step is to add the contents of location 204, the high byte of the multiplication to the new value of RAMTOP, which is itself in effect a high byte.

```
UA 10 REM ***************************
XY 20 REM *   CHARACTER SET COPIER   *
XZ 30 REM *      AND REDEFINER       *
AJ 40 REM *   MACHINE CODE ROUTINE   *
UR 50 REM * CHARACTER STRING WRITER  *
YM 60 REM * BY JOHN FOSKETT   FEB'96 *
UG 70 REM ***************************
FJ 80 GRAPHICS 0:POKE 16,64:POKE 53774,64
   :DIM F$(14):F$="D:STRING.LST"
KG 90 POKE 752,1:? :? "CHARACTER STRING W
   RITER":? :? "PRESS START TO BEGIN"
CM 100 ON PEEK(53279)<>6 GOTO 100:? :? "W
   RITING TO DISK - PLEASE WAIT":RESTORE
   :OPEN #1,8,0,F$
WO 110 REM  Write MC String
QI 120 ? #1;"110CHSET$=";CHR$(34);
BN 130 READ J:IF J<0 THEN 160
YZ 140 ? #1;CHR$(J);:GOTO 130
BU 150 REM  End
HB 160 ? #1;CHR$(34):CLOSE #1:? :? "FILE
   WRITTEN IN THE LIST FORMAT":? :? "FILE
    NAME: ";F$(3);"[ESC,BELL]":END
EG 170 REM  Machine Code String Data
KP 180 DATA 169,0,168,133,203,133,205,133
   ,213,162,4,173,244,2,133,204,165,106,1
   33,206,177,203,145,205,200,208,249
HE 190 DATA 230,204,230,206,202,208,242,1
   04,104,104,133,203,133,204,6,203,6,203
   ,6,203,162,5,70,204,202,208,251,165
WU 200 DATA 204,24,101,106,133,204,104,13
   3,206,104,133,205,104,104,133,207,177,
   205,145,203,200,196,207,208,247,165
PH 210 DATA 106,133,212,141,244,2,96,-1
```

Underline = INVERSE CHARACTERS · [ ] = CONTROL + CHARACTER · < > = INVERSE CONTROL + CHARACTER

The above calculations allows access to the first byte of the first character of the group to be redefined. With this address (locations 203/204) and the address of the redefining string (CHAR$) (locations 205/206), it is a simple matter to copy from one to the other within a loop (LOOP3). The loop will continue to copy the redefining string into the selected part of the new character set until the value stored in location 207 (the length of the redefining string) is met after which the loop is exited.

The last action achieved by the routine is to point CHBAS (location 756) to the new character set (that is RAMTOP) and to return the value of RAMTOP itself to BASIC via location 212 (location 213 being previously zeroed) should it be required.

THE DEMONSTRATION

The actual machine code routine is the string immediately below the REM header and the rest of the listing is the demonstration program.

The demonstration program redefines all 26 of the lower case characters which are displayed on the screen when the program is run in various patterns to show how they might be used. Note that when the program is run, the character set is copied into RAM and the stated characters are redefined instantly. You may find some, if not all of the of the redefined characters useful for your own purposes and of course it is up to you how many and which characters you wish to redefine.

In the demonstration program, the characters are redefined as follows....

- a: Pound sign
- b: Curved apostrophe (alternative to Shift-7)
- c: Curved opening quotes (alternative to Shift-2)
- d: Curved closing quotes (alternative to Shift-2)
- e: Underline (opposite of underscore)
- f: Double underline
- g: Curved left hand end block (for inverse characters)
- h: curved right hand end block (for inverse characters)
- i: Left pointing arrow head
- j: Left pointing arrow tail
- k: Right pointing arrow tail
- l: Right pointing arrow head
- m: Diamond
- n: Square
- o: Curved top-left corner (alternative to Control-Q)
- p: Curved top-right corner (alternative to Control-E)
- q: Curved bottom-left corner (alternative to Control-Z)
- r: Curved bottom-right corner (alternative to Control-C)
- s: Top-Left corner) Used for completing
- t: Top-right corner) corners when using
- u: Bottom-left corner) Control-B/M/N/V
- v: Bottom-right corner) characters
- w: Large chequer pattern
- x: Small chequer pattern
- y: Top-right to bottom-left diagonal cross hatch pattern
- z: Top-left to bottom-right diagonal cross hatch pattern

REDEFINING THE CHARACTERS

There have been many articles published over the years which describes exactly how to redefine the character set and redefining characters for this routine is achieved in exactly the same way.

To briefly outline the procedure by example, consider the first redefined character in the


```
PM 10 REM ***************************
WV 20 REM *     UTILITY PROGRAM      *
OC 30 REM *    FOR WRITING CHAR$     *
AI 40 REM *         FOR THE          *
DG 50 REM *  CHARACTER SET COPIER    *
GQ 60 REM *      AND REDEFINER       *
AU 70 REM * BY JOHN FOSKETT  FEB'96 *
PT 80 REM ***************************
QT 90 GRAPHICS 0:POKE 16,64:POKE 53774,64
   :RESTORE :POKE 752,1
XB 100 ? " CHARACTER SET COPIER AND REDEF
   INER":? :? " UTILITY PROGRAM  FOR WRIT
   ING CHAR$"
JI 110 ? :? :? "[ESC,TAB][ESC,TAB]WRITTEN
    BY":? "[ESC,TAB][ESC,TAB][ESC,LEFT]JO
   HN FOSKETT":? :? :? :? "[ESC,TAB]PRESS
    START TO WRITE CHAR$"
MK 120 ON PEEK(53279)<>6 GOTO 120:? "
   [ESC,UP]WRITING CHAR$ TO DISK --
    PLEASE WAIT":LINE=190:OPEN #1,8,0,"D:
   CHARSTR.LST"
XK 130 REM  Write CHAR$
VQ 140 I=1:? #1;LINE;"CHAR$=";CHR$(34);
EO 150 READ J:IF J<0 THEN 190
RX 160 IF I=97 OR I=193 THEN ? #1;CHR$(34
   ):LINE=LINE+10:? #1;LINE;"CHAR$(";I;")
   =";CHR$(34);
RW 170 ? #1;CHR$(J);:I=I+1:GOTO 150
CA 180 REM  End
DW 190 ? #1;CHR$(34):CLOSE #1
JM 200 POKE 752,0:? "[ESC,UP] CHAR$ WRITT
   EN IN LIST FORMAT USING ":? "[ESC,TAB]
     FILE NAME: CHARSTR.LST[ESC,BELL]":EN
   D
KW 210 REM ----------------------------
GQ 220 REM   CHARACTER REDEFINING DATA
PC 230 REM     32 CHARACTERS MAXIMUM
LC 240 REM ----------------------------
IW 250 REM  (1) Pound Sign
OD 260 DATA 28,54,48,120,48,48,126,0
BR 270 REM  (2) Apostrophie
WV 280 DATA 12,12,24,0,0,0,0,0
DW 290 REM  (3) Opening Quotes
ZV 300 DATA 204,204,102,0,0,0,0,0
EJ 310 REM  (4) Closing Quotes
WB 320 DATA 51,51,102,0,0,0,0,0
UA 330 REM  (5) Underline
ZC 340 DATA 0,255,0,0,0,0,0,0
JB 350 REM  (6) Double Underline
EL 360 DATA 0,255,0,255,0,0,0,0
HJ 370 REM  (7) Left End Block
CN 380 DATA 1,3,7,7,7,7,3,1
CQ 390 REM  (8) Right End Block
RS 400 DATA 128,192,224,224,224,224,192,1
   28
IQ 410 REM  (9) Left Arrow Head
BX 420 DATA 3,15,63,255,255,63,15,3
WG 430 REM  (10) Left Arrow
DM 440 DATA 51,102,204,255,255,204,102,51
LX 450 REM  (11) Right Arrow Tail
DQ 460 DATA 204,102,51,255,255,51,102,204
MF 470 REM  (12) Right Arrow Head
RO 480 DATA 192,240,252,255,255,252,240,1
   92
EF 490 REM  (13) Diamond
EN 500 DATA 24,60,126,255,255,126,60,24
JK 510 REM  (14) Square
UX 520 DATA 255,255,195,195,195,195,255,2
   55
YJ 530 REM  (15) Curved Corner T-Left
UQ 540 DATA 0,0,0,7,15,28,24,24
TG 550 REM  (16) Curved Corner T-Right
IB 560 DATA 0,0,0,224,240,56,24,24
FN 570 REM  (17) Curved Corner B-Left
LY 580 DATA 24,24,28,15,7,0,0,0
AK 590 REM  (18) Curved Corner B-Right
WA 600 DATA 24,24,56,240,224,0,0,0
JU 610 REM  (19) Corner Top-Left
JY 620 DATA 0,0,0,0,0,0,3,3
FT 630 REM  (20) Corner Top-Right
PI 640 DATA 0,0,0,0,0,0,192,192
```

continued

Underline = INVERSE CHARACTERS · [ ] = CONTROL + CHARACTER · < > = INVERSE CONTROL + CHARACTER

demonstration program, the pound sign....

BYTE NUMBER	CHARACTER PATTERN	CALCULATED VALUE OF BITS	DATA VALUE
Byte 1	00011100	16+8+4	= 28
Byte 2	00110110	32+16+4+2	= 54
Byte 3	00110000	32+16	= 48
Byte 4	01111000	64+32+16+8	= 120
Byte 5	00110000	32+16	= 48
Byte 6	00110000	32+16	= 48
Byte 7	01111110	64+32+16+8+4+2	=126
Byte 8	00000000	0	= 0

Therefore the DATA for the pound sign would be....

DATA 28,54,48,120,48,48,126,0

Note that the above is the same as the first DATA line (the pound sign) in the utility program.

Because this routine redefines up to 32 characters instantly, at machine code speed, a character string is required to represent the DATA obtained from the above procedure and this is achieved in the demonstration program using CHAR$.

THE UTILITY PROGRAM

With experience, the string can be constructed directly by hand, but it is far more convenient to use a separate utility program to write the string for you to disk from the calculated DATA. To write the string, simply run the program and press START after which the string will be written to disk using the file name "CHARSTR.LST" in the LIST format. Once written to disk, all you then need to do is to ENTER the string into your program.

The utility program is prepared for writing CHAR$ as it appears in the demonstration program. When redefining your own characters, it is a simple matter to replace the DATA statements at the end of the listing with your own DATA. The first line number used for CHAR$ is defined on line 120 which may need to be changed to suit your own needs.

TYPING THE CHARACTER STRINGS

As described above, the utility program will write CHAR$ for you and a third program has been included to write the machine code string itself from DATA. Run the program and press START to write the string to disk using the file name "STRING.LST" in the LIST format. Once written, the string can then simply be ENTERed into the program as before.

```
AC 650 REM (21) Corner Bottom-Left
HM 660 DATA 3,3,0,0,0,0,0,0
CH 670 REM (22) Corner Bottom-Right
AW 680 DATA 192,192,0,0,0,0,0,0
RO 690 REM (23) Large Chequer
UF 700 DATA 195,195,60,60,60,60,195,195
AI 710 REM (24) Small Chequer
OR 720 DATA 153,102,102,153,153,102,102,1
   53
XG 730 REM (25) Cross Hatch ///
WL 740 DATA 204,153,51,102,204,153,51,102
NG 750 REM (26) Cross Hatch \\\
WB 760 DATA 51,153,204,102,51,153,204,102
VP 770 REM End of DATA flag
GS 780 DATA -1
```

Underline = INVERSE CHARACTERS · [ ] = CONTROL + CHARACTER · < > = INVERSE CONTROL + CHARACTER

DESERT ISLAND DISKS

Strolling round our local John Menzies shop recently, I came across a new Atari magazine (ST only I'm afraid), called Atari world. It was a refreshing change from the usual PC this and Mac that and, since I'd recently acquired my first ST, I decided to take the plunge.

Reading through the magazine I found an article that I thought might be ideal for Page 6. The idea was that users of the ST were to decide which three items of software they would like to have with them if they were marooned on a desert Island. Of course, a little bit of imagination would be needed since this Island would have to have all the comforts of home! Still it got me thinking, so here follows my selection of Atari Classic Faves.

I bought my first Atari 800XL and 1050 Disk drive in 1988 in a sell-off package from Curry's for the princely sum of £120. The bundled in software package comprised the drawing program Paint, the adventure game The Pay-Off and the first of my choices, **THE HOME FILING MANAGER**. This easy to use database has been in use almost since the day I first tried it. It takes the form of a card index system, allowing you to flick through each record easily. Also included are search, edit, copy and printing functions. It's many uses include the good old address and contact list and listing the contents of all my Atari user and Page 6 magazines.

My second choice is that Jack of all trades, **MINI OFFICE II**. This bargain priced piece of software is a combination of 6 different utilities comprising a word processor, spreadsheet, database, graphics display, label printer and communications modules. In fact this article has been prepared using the word processor package. The spreadsheet module has been an absolute Godsend in getting our finances back on track, while the database module has found use in organising my catalogue of software. As yet I haven't used the other modules (after all I don't need a graph to tell me I'm broke!) but the label printer and comms packages may yet find use.

My third and final choice is an extremely useful utility entitled **VIDEO TITLE SHOP**. The complete, 5 disk, package comprises the Video Title Shop program itself, the MicroPainter art package and the Graphics Companions 1 and 2. The many uses of the complete package go far beyond that of putting titles onto videos. The graphics companions themselves contain many fonts, border styles and clip art that may be used in text processors and art programs and while the MicroPainter art package does leave something to be desired, the complete package is unrivalled when used in it's primary role of video titling. It has certainly given a lot more fizz to my home "Video Nasties".

There are many excellent programs that I have not included as one of my 3 choices. Certainly, I do use more software than I've listed here. If you disagree with my choices why not write in and tell us of your 3 favourites. After all, variety is the spice of life, especially on a desert Island!

This article seems to be anonymous as there was no name on the disk. If it has become separated from a covering letter we apologise. Thanks, whoever you are. If you would like to own up we'll credit you next issue.

The Desert Island Disks idea seems to be a great one, and one that is easy for almost anyone to write. Think about your three favourite programs of all time and write about them and we'll get a regular Desert Island column going that will let everyone know what are the most respected programs of all time.



TUTORIAL

TEXTPRO MACROS PART 2

Frank Walters continues his exploration of the macro facility of the Atari's best PD word processor

There are nine <inverse> 'CONTROL' characters that have special functions when used in a macro. On your screen they look like funny block characters. Each special macro command must be entered in the ESCape mode. [SELECT] must then be pressed while typing [CONTROL] then the desired letter. The key strokes would be as follows:

[ESC] [SELECT]_[CTRL]_[desired letter]

After you press all those keys, only one funny looking character will appear on the screen. So, while all the examples in this instalment look long, they don't really take up much screen space once typed.

In the last part of this series we used one of these special macro commands - [ESC]<[CONTROL]_[G]> to 'GoTo' or link one macro to another. Now I will explain the other eight special macro commands: A, E, I, K, M, P, R and Y in more detail.

ASK PROMPT

[ESC]<[CONTROL]_[A]>

[ESC]<[CONTROL]_[A]> should be followed by the desired prompt which is then terminated by a [RETURN]. The 'Ask' prompt requires the user to reply with 'Yes' or 'No'. The default is 'No' so if [Y] is not pressed then No is assumed. If 'Yes' is the reply, then the macro continues with whatever you wish it to do, which is whatever keys follow the [RETURN] character after the 'Ask' prompt line. If No is the reply, TextPRO will look for the macro defined by the [&] key and will execute it. If [&] is not defined as a macro, then the macro will end after a 'No' response.

The 'Ask' prompt will display your prompt in the status line at the top of the screen. It will automatically add the following message to your text:

: Sure? (Y/N)

Note that a colon immediately follows your question, in place of a carriage return. Here is an example macro using the <[CONTROL]_A> Ask prompt:

#<=>[ESC]<[CONTROL]_[A]>XClear Screen?[RETURN]
[ESC][SHIFT]_[CLEAR][Y]&<=>The screen didn't clear!

The [#] is the [START] key macro. Now, load this macro with [CONTROL]_[V] and press [START]. What you will see in the status line is the following:

Clear Screen? : Sure? (Y/N)

If your reply is 'Yes' the macro will execute the 'clear screen' command on the next line of the macro. Remember, you need to be in 'Escape Mode' to send the 'clear screen' command. When you type [ESC][SHIFT]_[CLEAR] you will only see one character that looks like an arrow pointing up and to the left. The [Y] after [ESC][SHIFT]_[CLEAR] answers the question you normally get when you try to clear the screen from the editor (ERASE ALL TEXT? : Sure (Y/N)). Note that there is no [RETURN] between the [Y] and the &, otherwise that carriage return would print on the screen after it was cleared.

If the response is 'No' for the above macro, the [&] macro is executed and the message "The screen didn't clear!" will be typed on the screen. Sometimes you will want to send a message to the status line (not the screen) to indicate how to restart the macro. This message would be done with the special [ESC]<[CONTROL]>_[Y] key described later.

ERASE FILE

[ESC]<[CONTROL]_[E]>

Simply follow the [ESC]<[CONTROL]_[E]> with the device and filename you wish to erase. For example:

[ESC]<[CONTROL]_[E]>D1:DUMMY.TXT [RETURN]

The filename must be followed by a [RETURN]. You might want to use the Erase File function in conjunction with the Input command ([ESC]<[CONTROL]>_[I]>) which pauses the macro to permit the user to type the filename to erase. [ESC]<[CONTROL]>_[I]> is described later.

GOTO ANOTHER MACRO

[ESC]<[CONTROL]>_[G]>

In the following example the macro ends with the GoTo command, followed by the macro key you wish to execute next. When defining a letter macro key, I normally define both upper case and lower case letters to the

same macro so it doesn't matter which case is selected by the user. The [i] macro must be defined somewhere else in the macro file.

I <=>[ESC]<[CONTROL]_[G]>i (no RETURN added)

This is a somewhat trivial but powerful example. Both [OPTION]_[I] and [OPTION]_[i] will execute the same macro (once [i] is defined). This special macro command was more fully explained in the first part of the series last issue.

INPUT MODE

[ESC]<[CONTROL]>_[I]>

This is a powerful mode. It permits the macro to pause for user input, similar to INPUT in Basic. When [ESC]<[CONTROL]_[I]> appears in a macro, it may either be preceded by a prompt in the status line to explain what is requested, or it might follow a [CONTROL]_[S(ave)] or [CONTROL]_[L(oad)] command which displays the prompt automatically.

The user types the information and ends the Input Mode with a [RETURN]. But like Basic, the Return is not part of the Input and will not be sent to the editor unless your macro has a [RETURN] character following the [ESC]<[CONTROL]>_[I]>. During Input Mode, the cursor keys are suppressed and only a limited number of TextPRO 'CONTROL' characters are available to the user until Input Mode is terminated. Here is an example of the Input Mode that I use to enter printer codes for italics 'on' and 'off' while typing in the editor. I've defined the printer codes for italics as inverse <I> for 'on' and inverse <J> for 'off'. I use the [OPTION]_[I] or [OPTION]_[i] as the macro for italics:

I<=>[ESC]<[CONTROL]_G>ii<=><I>[ESC]<[CONTROL]_[I]><J> (no return needed)

Invoking the macro types inverse <I> into the editor to turn italics 'on'. Then the Input Mode is entered. The user continues typing and the characters are printed to the editor as they are typed. Backspace/delete must be used for corrections, since cursor keys are not available in the Input Mode. When the user decides to end the italics printing he presses [RETURN] and the inverse <J> is typed automatically and the Input Mode ends. With no [RETURN] following the [ESC]<[CONTROL]_[I]> in the macro, the user's [RETURN] is not sent to the editor, but is simply an indicator to the macro to end the Input Mode and continue executing to the end of the macro by typing an inverse <J> into the editor.

KEYPRESS WAIT

[ESC]<[CONTROL]_[K]>

[ESC]<[CONTROL]_[K]> in a macro pauses the macro until the user presses any key. The key will not be sent to the editor, but the macro will resume with whatever follows the [ESC]<[CONTROL]_[K]> in the current macro key. It can be used to pause at a message in the status line to inform the user of something. The "PRESS ANY KEY" message is not sent automatically when using [ESC]<[CONTROL]_[K]> but you can print a message there with a [ESC]<[CONTROL]_[Y]> command immediately preceding the [ESC]<[CONTROL]_[K]>.

MENU DIRECTORY BRANCH

[ESC]<[CONTROL]_[M]>

This is the TextPRO macro equivalent of the GOSUB command in Basic. It is a bit complicated so I will leave it out here and come back to it later in the article.

PRE-SELECT MACRO

[ESC]<[CONTROL]_[P]>

[ESC]<[CONTROL]_[P]> is used to designate any macro key to execute automatically after another macro is loaded via the [CONTROL]_[V] command within the macro. The syntax requires that [ESC]<[CONTROL]_[P]> be followed by the macro key to be executed (after the next macro is loaded) and then the [CONTROL]_[V] command with the device and filename of the macro to be loaded from disk:

?<=>[ESC]<[CONTROL]_[P]>@[CONTROL]_[V]TEXTPRO.MAX[RETURN]

Here the [HELP] key [?] is designated as a macro key to pre-select the [@] (automatic) macro key after TEXTPRO.MAX loads. [CONTROL]_[V] is the "Load Macro" command. TEXTPRO.MAX is the filename of the macro to be loaded and must be terminated with a [RETURN]. When TEXTPRO.MAX is loaded, if [@] is defined as a macro elsewhere, it will execute automatically. Any macro key can be used following the [ESC]<[CONTROL]_[P]>, not just the [@], but for simplicity and standardisation, the [@] key is normally used as the automatic macro key. This is the key TEXTPRO.MAX designates when it loads another disk macro file via the [START] key.

RENAME A FILE

[ESC]<[CONTROL]_[R]>

Here's the syntax:

[ESC]<[CONTROL]_[R]>D8:OLDNAME.TXT, NEWNAME.TXT[RETURN]

Just like using Atari DOS, include the drive number for the old filename, so TextPRO can find it. Just the new filename is required following the comma. This special macro command must be terminated by a [RETURN].

PRINT MESSAGE

[ESC]<[CONTROL]_[Y]>

This must be followed by your message and terminated by a [RETURN]. Your message, all characters up to the [RETURN], will be printed on the status line until the next keypress. This is one of the most useful of all special keys as you keep the user informed as to what to do next, but without messing up what is in the editor itself.

Referring back to a previous example, I use the [ESC]<[CONTROL]_[Y]> Print Message to status line to indicate italics 'on' mode. I insert it immediately before the [ESC]<[CONTROL]_[I]> in the macro. I also add another [ESC]<[CONTROL]_[Y]> following the <J> to remind the user when italics is 'off'. Here's what that looks like when added to my previous example:

I<=>[ESC]<[CONTROL]_[G]>ii<=><I>[ESC]<[CONTROL]_[Y]>Italics On[RETURN]

[ESC]<[CONTROL]_[I]><J>[ESC]<[CONTROL]_[Y]>Italics Off[RETURN]

LET'S TACKLE THE HARD ONE!

That just about concludes the special macro keys but, if you recall, we said we would come back to one. The permanent message in the status line is no longer available for use in macros after version 4.55 as that special macro command [ESC]<[CONTROL]_[M]> is now used for the Menu directory branch (GOSUB) function. This macro is very powerful (and confusing!) so let's tackle it now.

I had more trouble figuring out how to use this special macro key than any other function of macros. It can be tricky and you can get yourself lost as you sometimes do when using GOSUB with IF/THEN statements in Basic. If you don't program Basic, it could be even more confusing. I still get lost sometimes with the Menu Branch.

WHAT IS THIS KEY?

[ESC]<[CONTROL]_[M]> selects the Menu Branch function within a macro set. To review the keyboardology you must press [ESC] first, then hold [SELECT] for inverse, then press [CONTROL]_[M] to type the character. It will look like a small white box on your screen.

This is a very powerful macro key that allows the user to optionally branch to any number of macro keys available by just pressing a single key without [OPTION]. When the Menu Branch is invoked, if a key is pressed that has no defined macro key (an illegal keypress), then the current macro simple stops after the keypress and no branch takes place. If the next item following the [ESC]<[CONTROL]_[M]> is the definition of another macro, then the macro simple stops after an illegal keypress which may not be desirable. Also, when one of the selected macro keys is pressed after the Menu Branch command, once the selected macro key is executed, the branch returns to the next key *AFTER* the Menu Branch command and whatever (if anything) follows the Menu Branch is executed at that time. This is what can get confusing when using this function.

Think of it like GOSUB and RETURN in Basic. [ESC]<[CONTROL]_[M]> is the GOSUB. The user picks the "line number", in this case which macro key to GOSUB to. At the end of that macro, it RETURNs to the statement (or character) following the [ESC]<[CONTROL]_[M]> special command and continues from there.

HOW DO I USE THE MENU BRANCH?

[ESC]<[CONTROL]_[M]> will automatically print to the status line whatever you type following the Menu Branch key. Your "prompt" must end with a [RETURN]. Normally you list the macro keys you have available to choose from the menu. I recommend using inverse characters to highlight the available macro keys. Number keys are easy to use and simpler to define as macros since there is no upper/lower case to worry about. Look at the following example:

#<=>[ESC]<[CONTROL]_[M]><1>-Print <2>-Skip <3>-Repeat <4>-Quit[RETURN]

The first thing we must think about is what happens if 1, 2, 3, or 4 are not pressed. One way to "trap" an illegal key is to continue following the Menu Branch with a 'GoTo' back to the Menu Branch macro function, as shown below:

[ESC]<[CONTROL]_[G]># (no Return needed)

So any key other than a defined macro key will go back to the prompt again. (See CAUTION later in the article, though).

Also, after executing 1, 2, 3 or 4 the Menu will be displayed again because TextPRO will RETURN from any of the defined keys to the GoTo statement. When this technique is used, you must include an option to quit or exit from the loop. [HELP] (?) is usually defined to exit back to TEXTPRO.MAX and in the above example you can link macro key 4 to 'GoTo' the [HELP] or (?) key as follows:

4<=><CTRL G>??<=>[CTRL_V]TEXTPRO.MAX

"4" could load TEXTPRO.MAX directly, bypassing the use of ?, but for standardisation of all my disk macro files, I always define ? to reload TEXTPRO.MAX. Besides, the Menu Branch may only be a small part of this particular macro and the HELP key might be used to terminate at another point in the macro. You would now go on to define 1, 2 and 3 keys for whatever you intended for this macro.

CAUTION: Using the "GoTo" back to the Menu Branch macro immediately after the menu prompt makes it almost impossible to break out of the loop except by loading another macro file from disk or hitting the Break key, which is messy. So don't use this technique if you want to have other functions besides the Menu Branch in your macro set. Having the [START] (#) key define the Menu Branch makes it easy to get back into the loop if you hit an illegal key to end the loop by mistake.

I'M CONFUSED ALREADY!

Are the keys shown in the menu the only ones you can choose? No. You show the macro keys you want the user to pick from, but all macro keys defined in the macro file are legal keys and can be used from the Menu Branch prompt. They would execute and return to the point following the [ESC]<[CONTROL]_[M]> statement.

ANOTHER EXAMPLE

Type in the example shown in the box overleaf and save it as CR.MAX so we can go through it step by step.

CR.MAX is a macro I made to selectively remove Carriage Returns (C/R) and replace them with a space. I use it to reformat messages captured in 80 columns where I don't want the Returns removed from the headers or between paragraphs. It doesn't use the "GoTo" loop back to the Menu Branch command so I can break out easily to edit and then return to the Menu with [START]. I begin by defining the macro name with <i> so it can be printed to the prompt line. If the previous macro pre-selects <i> for execution, then the name of CR.MAX will be displayed in the status line after it loads. I then define the [HELP] key (?) to re-load the default macro. Notice that I also pre-select <i> to execute prior to loading TEXTPRO.MAX. That way after TEXTPRO.MAX loads, the <i> macro in it will execute, loading the name of the macro (TEXTPRO.MAX) into the prompt line. Using <i> in this way with pre-selects is pretty

CR.MAX

```
<i><=>[ESC]<[CONTROL]_[Y]> Active Macro: CR.MAX[RETURN]
?<=>[ESC]<[CONTROL]_[P]><i>[CONTROL]_[V]TEXTPRO.MAX[RETURN]
@<=>[ESC]<[CONTROL]_[Y]> Press START to replace C/R[RETURN]
#<=>[ESC]<[CONTROL]_[F]>[ESC][ESC]<[ESC]>[RETURN]
[ESC]<[CONTROL]_[C]>[SPACE][RETURN]
[ESC]<[CONTROL]_[G]>f
f<=>[CONTROL]_[F][ESC]<[CONTROL]_[M]> <Y>es <N>o <D>own page <Q>uit[RETURN]
[ESC]<[CONTROL]_[G]>@
Y<=><[CONTROL]_[G]>yy<=>[CONTROL]_[C]<[CONTROL]_[G]>f
N<=><[CONTROL]_[G]>nn<=>[CONTROL]_[F][CONTROL]_[F]<[CONTROL]_[G]>f
D<=>[CONTROL]_[G]dd<=>[ESC][CONTROL]_[=] (x 18) [CONTROL]_[H]<[CONTROL]_[G]>f
```

Note: (x18) means repeat [ESC][CONTROL]_[=] 18 times

much a TextPRO macro "standard".

<i><=>[ESC]<[CONTROL]_[Y]> Active Macro: CR.MAX[RETURN]

?<=>[ESC]<[CONTROL]_[P]><i>[CONTROL]_[V]TEXTPRO.MAX[RETURN]

Next I define the automatic macro to display an appropriate prompt.

@<=>[ESC]<[CONTROL]_[Y]> Press START to replace C/R[RETURN]

After the [START] key is pressed I "select find string" as [RETURN] and "select change string" as [SPACE].

#<=>[ESC]<[CONTROL]_[F]>[ESC][ESC]<[ESC]>[RETURN]

[ESC]<[CONTROL]_[C]>[SPACE][RETURN]

Getting that [RETURN] character into the command line is a complicated process as you can see from the keypresses required above. On the screen you will see a regular Escape character followed by an inverse Escape character.

Now I'm going to tell the macro to go to a "dummy" macro key, "f" (for find). This way I can loop back into the main "Find" section of the macro without having to redefine the "select find string" and "select change string" each time I move to another C/R.

[ESC]<[CONTROL]_[G]>f

Now comes the Menu Branch command. First we execute the "find" then display the options as to what to do with the C/R we found.

f<=>[CONTROL]_[F][ESC]<[CONTROL]_[M]> <Y>es <N>o <D>own page <Q>uit[RETURN]

<Y>, <N>, <D> and <Q> represent inverse

characters once again.

The next line tells the macro simply to display our starting prompt in case of an illegal keypress. That is, we execute the automatic macro.

[ESC]<[CONTROL]_[G]>@

Now I define the <Y>es key for both upper and lower case, do my "replace found string" and then go back to the beginning of the main "Find" macro, the dummy "f".

Y<=><[CONTROL]_[G]>yy<=>[CONTROL]_[C]<[CONTROL]_[G]>f

The <N>o macro key follows immediately after the "f" above since there is no [RETURN]. Note that we define upper and lower case again below:

N<=><[CONTROL]_[G]>nn<=>[CONTROL]_[F] [CONTROL]_[F]<[CONTROL]_[G]>f

The Find function is funny. When you use it successively, it goes from one C/R to the next. But in the macros, you need to send the [CONTROL]_[F] twice, since the first one finds the C/R your cursor is sitting on. The second [CONTROL]_[F] moves to the next one. In the case of <Y>es, the C/R is replaced by a space so you don't want the extra [CONTROL]_[F]. It is sort of like using [CONTROL]_[F] in the editor, then pressing another key, like [CONTROL]_[U], and pressing [CONTROL]_[F] again. After the [CONTROL]_[U], the "find string" function looks starting at the cursor position, since it forgot you already found that character before. Try it and you'll see what I mean. Note that we still loop back to the main "find" macro (f).

Now for the <D>own Page. This is useful when your cursor gets to the bottom of the screen and you can't see below it to tell if the C/R picked is at the end of a paragraph or not. Best not to scroll too far or you may scroll some lines off the top. That's what the (x18) is all about. Do the [ESC][CONTROL]_[=] series of key strokes 18 times (or copy them). I then "Home" cursor to top of screen and begin "Find again as shown below:

D<=>[CONTROL]_[G]dd<=>[ESC][CONTROL]_[=] (x 18) [CONTROL]_[H]<[CONTROL]_[G]>f

Finally we come to the <Q>uit. Since we may only want to edit something and not reload the default macro, I just don't define the <Q>uit so it goes to the automatic macro (@) message following the Menu Branch command, just like an illegal key.

The problem I have with this macro set is that it uses a lot of Menu Branches whenever one answers <Y> or <N> to remove or skip to the next [RETURN] character. Each time, the <Y> or <N> macro uses a GoTo back to the # [START] macro and the macro return is not executed. Eventually, when you <Q>uit or hit an illegal key, all those returns are executed in succession, flashing the Press START message on the status line until the same number of returns are executed as the number of keys I had pressed from the menu. Oh well, I just hit [BREAK] and press on until I can figure out a better way.

ARE THERE ANY OTHER MACRO KEYS?

In this article I have covered all 9 special macro keys, however there is one other key that is primarily for macro use but can also be used from the editor. It is not an inverse CONTROL character. I am talking about [CONTROL]_[T].

TURBO TIPS

by Robert De Letter

Here is a useful programming hint that will save memory and make for more compact programs when using Turbo Basic.

Firstly when you want to set the colour registers you would usually use, for example:

POKE 710, 60: POKE 709,42:POKE 710,45:POKE 711,50:POKE 712,88

but you can use instead:

MOVE ADR("<*-2X"),708,5

The letters in the brackets have the ATASCII values of the numbers POKEd into the registers in the previous example. To discover the letters/symbols to use simply type ? CHR$(60), for example and the character will be shown on screen.

Look out for another useful hint from Robert next issue!

There have to be many more great tips like these in your programming notebook so share them with others. Send in your Turbo Tips!

Try [CONTROL]_[T] from the keyboard and you will see the prompt:

Set [M,R,I,T,A,U,L]: RETURN to exit

Used in a macro, you can set certain 'toggle' modes to ensure your macro commands are going to function correctly.

M - Selects Main bank. If you want to ensure you are in a specific bank, then use [CONTROL]_[T] once followed by [M] the [SELECT]_[CONTROL]_[B] to switch to Bank 1, or add a [CONTROL]_[B] to switch from bank 1 to 2.

R, I - [CONTROL]_[I] Replace, Insert mode

T, A - [Atari Logo key] Text, Atascii mode

U, L - Upper, Lower case

WHAT KEYS CAN'T BE USED AS MACROS?

I can't think of any keys that could not, theoretically, be defined as macro keys by using the inverse <=> key. I thought maybe the <=> itself might be difficult but I tried it and it works. There is a problem if the <=> is defining itself unless it is the very first character in the macro file.

The <=> acts as a delimiter that defines the end of the previous macro. When a macro is executing, it is also looking ahead and when it sees a <=> it will stop executing when it reaches the character preceding the <=>. With two <=> in a row, the first <=> will prevent the previous macro from executing the last character in the macro because it thinks that character is being defined by the

continued on page 50



PROGRAMMING

GRAPHICS USING MACHINE CODE

by Peter Foote

Reading James Mathrick's letter in issue 76 of NAU brought to mind a similar program I had previously created - see NAU issue 57. The problem James posed was how to change the graphics modes using machine code. Here is a quick guide.

FIRST IN BASIC

To produce a screen in BASIC you can use either;

GRAPHICS NOTXT+NOCLR+MODE

Where NOTXT = 0 for a text window, 16 no text window, NOCLR = 0 clear screen, 32 no clear screen, MODE = (0 to 15) screen format.

or

CLOSE #CHANNEL
OPEN #CHANNEL,READ+WRITE+TEXT+NOCLR,MODE,"S:"

Where CHANNEL = channel number (1 to 7), READ = 0 no read, 4 read from screen, WRITE = 8 write to screen, TEXT = 0 no text window, 16 text window

continued on page 50

```
DP 10 REM DEMO  -  TRIANGLE
XI 11 REM BY P.A.FOOTE 1996
EH 20 CLOSE #6:REM GRAPHICS 3
YR 30 OPEN #6,28,3,"S:"
UR 40 REM WRITE IN TEXT WINDOW
HT 50 POKE 656,1:POKE 657,17:POKE 658,0
TJ 60 PRINT "ATARI"
GK 70 POKE 656,2:POKE 657,15:POKE 658,0
PH 80 PRINT "COMPUTERS"
SU 90 REM DRAW TRIANGLE
XJ 100 COLOR 1
UT 110 PLOT 5,17:REM DRAW 2 SIDES
PI 120 DRAWTO 35,17
FE 130 DRAWTO 20,2
AI 140 POKE 765,2:REM FILL COLOUR
TM 150 POSITION 5,17
UP 160 XIO 18,#6,4,0,"S:"
JJ 170 REM CYCLE PIXEL
SC 180 A=PEEK(20)+100:REM WAIT
JP 190 IF A>255 THEN A=A-256
NL 200 IF PEEK(20)<>A THEN GOTO 200
VY 210 LOCATE 20,10,C
OZ 220 COLOR C+1:PLOT 20,10
OM 230 GOTO 170
```

Underline = INVERSE CHARACTERS - [ ] = CONTROL + CHARACTER - < > = INVERSE CONTROL + CHARACTER

```
0010 ; GRAPHICS DEMO
0020 ;BY P.A.FOOTE 1996
0030 ;
0040 RTCLOK =$14 REAL TIME CLOCK
0050 ROWCRS =$54 CURSOR ROW
0060 COLCRS =$55 CURSOR COLUMN
0070 OLDROW =$5A OLD ROW
0080 OLDCOL =$5B OLD COLUMN
0090 IOCB =$0340 IOCB BASE ADDRESS
0100 TXTROW =$0290 TEXT CURSOR ROW
0110 TXTCOL =$0291 TEXT CURSOR
COLUMN
0120 ATACHR =$02FB CHARACTER/COLOUR
0130 FILDAT =$02FD FILL COLOUR
0140 CHANNEL =$0600 CHANNEL NUMBER
0150 GRMODE =$0601 GRAPHICS MODE
0160 GRTYPE =$0602 SCREEN FUNCTION
0170 CIOV =$E456 CIO VECTOR
0180 ;
0190 *=$8000 START ADDRESS
0200 ;
0210 START LDX #16*6 CHANNEL 6
0220 STX CHANNEL
0230 JSR CLOSE CLOSE #6
0240 LDA #3 GRAPHICS 3
0250 STA GRMODE
0260 LDA #28 (CLEAR SCREEN,
0270 STA GRTYPE TEXT WINDOW,
0280 JSR GROPEN & SCREEN READ)
0290 BMI SKIP BRANCH ON ERROR
0300 ;
0310 LDX #16*0 SELECT
0320 STX CHANNEL TEXT WINDOW
0330 LDX #17 POSITION 17,1
0340 LDY #1
0350 JSR TXTPOS
0360 LDY #0 PRINT
0370 JSR PRINT "ATARI"
0380 BMI ERROR OUT OF RANGE?
0390 LDX #15 POSITION 15,2
0400 LDY #2
0410 JSR TXTPOS
0420 LDY #4 PRINT
0430 JSR PRINT "COMPUTERS"
0440 BMI ERROR OUT OF RANGE?
0450 LDX #0 POSITION 0,0
0460 LDY #0
0470 JSR TXTPOS
0490 LDX #16*6 SELECT
0500 STX CHANNEL MAIN WINDOW
0510 LDA #1 COLOR 1
0520 JSR COLOR
0530 LDX #5 PLOT 5,17
0540 LDA #0
0550 LDY #17
0560 JSR PLOT
0570 SKIP BMI ERROR OUT OF RANGE?
0580 LDX #35 DRAWTO 35,17
0590 LDA #0
0600 LDY #17
0610 JSR DRAWTO
0620 BMI ERROR OUT OF RANGE?
0630 LDX #20 DRAWTO 20,2
0640 LDA #0
0650 LDY #2
0660 JSR DRAWTO
0670 BMI ERROR OUT OF RANGE?
0680 LDA #2 FILL COLOUR
0690 JSR FILCOL = 2
0700 LDX #5 FILL
0710 LDA #0 END AT 5,17
0720 LDY #17
0730 JSR FILL
0740 BMI ERROR OUT OF RANGE?
0750 ;
0760 LOOP LDA RTCLOK WAIT 2
0770 CLC SECONDS
0780 ADC #100
0790 WAIT CMP RTCLOK
0800 BNE WAIT
0810 LDX #20 LOCATE X,Y,C
0820 LDA #0 AT 20,10
0830 LDY #10
0840 JSR LOCATE
0850 BMI ERROR
0860 LDX ATACHR COLOUR = C+1
0870 INX
0880 TXA
0890 JSR COLOR
0900 LDX #20 PLOT 20,10
0910 LDA #0
0920 LDY #10
0930 JSR PLOT
0940 BMI ERROR OUT OF RANGE?
0950 JMP LOOP REPEAT LOOP
0960 ERROR BRK STOP
```



```
0970 ;
0980 TXTTAB .WORD TXT1,TXT2-TXT1
0990 .WORD TXT2,TXT3-TXT2
1000 ;
1010 TXT1 .BYTE "ATARI"
1020 TXT2 .BYTE "COMPUTERS"
1030 TXT3
1040 ;
1050 CLOSE LDA #12 CLOSE CHANNEL
1060 JMP EXECUTE
1070 ;
1080 GROPEN LDX CHANNEL GET CHANNEL
1090 LDA #3 SET COMMAND
1100 STA IOCB+2,X BYTE
1110 LDA #GRVEC&255 POINT TO
1120 STA IOCB+4,X HANDLER I.D.
1130 LDA #GRVEC/256
1140 STA IOCB+5,X
1150 LDA #2 SET LENGTH TO
1160 STA IOCB+8,X TWO BYTES
1170 LDA #0
1180 STA IOCB+9,X
1190 LDA GRTYPE GET TYPE OF
1200 STA IOCB+10,X SCREEN
1210 LDA GRMODE GET MODE
1220 STA IOCB+11,X
1230 JSR CIOV EXECUTE OPEN
1240 RTS EXIT
1250 GRVEC .BYTE "S:" SCREEN I.D.
1260 ;
1270 POSITION STY ROWCRS POSITION X,Y
1280 STX COLCRS
1290 STA COLCRS+1
1300 RTS
1310 ;
1320 TXTPOS LDA #0 POSITION X,Y
1330 STY TXTROW FOR TEXT
1340 STX TXTCOL WINDOW
1350 STA TXTCOL+1
1360 RTS
1370 ;
1380 PRINT LDX CHANNEL GET CHANNEL
1390 LDA #11 SET COMMAND
1400 STA IOCB+2,X FOR 'PUT'
1410 LDA TXTTAB,Y GET TEXT
1420 STA IOCB+4,X ADDRESS
1430 LDA TXTTAB+1,Y
1440 STA IOCB+5,X
1450 LDA TXTTAB+2,Y GET TEXT
1460 STA IOCB+8,X LENGTH
1470 LDA TXTTAB+3,Y
1480 STA IOCB+9,X
1490 JSR CIOV EXECUTE PRINT
1500 RTS EXIT
1510 ;
1520 COLOR STA ATACHR COLOUR 'A'
1530 RTS
1540 ;
1550 FILCOL STA FILDAT FILL COLOUR
1560 RTS
1570 ;
1580 PLOT INY PLOT X,Y
1590 STY OLDROW
1600 STX OLDCOL
1610 STA OLDCOL+1
1620 DEY
1630 ;
1640 DRAWTO JSR POSITION DRAWTO X,Y
1650 LDA #17
1660 JMP EXECUTE
1670 ;
1680 FILL JSR POSITION FILL
1690 LDA #18
1700 ;
1710 EXECUTE LDX CHANNEL DO
COMMAND
1720 STA IOCB+2,X
1730 JSR CIOV
1740 RTS
1750 ;
1760 LOCATE JSR POSITION GET PIXEL
1770 LDX CHANNEL COLOUR
1780 LDA #7 AT X,Y
1790 STA IOCB+2,X ANSWER IN
1800 LDA #ATACHR&255 ATACHR
1810 STA IOCB+4,X
1820 LDA #ATACHR/256
1830 STA IOCB+5,X
1840 LDA #1
1850 STA IOCB+8,X
1860 LDA #0
1870 STA IOCB+9,X
1880 JSR CIOV
1890 RTS
1900 ;
1910 .END END OF PROGRAM
```

NOCLR = 0 clear screen, 32 no clear screen and MODE = (0 to 15) screen format

or

```
CLOSE #CHANNEL
XIO 3,#CHANNEL,READ+WRITE+TEXT+
NOCLR,MODE,"S:"
```

XIO 3 is the same as the OPEN command. All variables are as shown in the example above.

BASIC uses channel 6 for most of its graphics commands, the exceptions being either a GRAPHICS 0 screen or the text window. The last two examples show that the Atari uses the input/output control blocks (IOCBs) to execute most of the graphics commands that are available in BASIC.

Once a screen is initiated you no doubt will wish to write graphics and text to it. Check out the BASIC program, PROGRAM 1, which draws a coloured triangle on a graphics 3 screen with a text window, then cycles the centre pixel through the four playfield colours.

NOW THE SAME THING IN MACHINE CODE

In machine code it would look like program 2. It looks quite frightening but you should be able to pick out the equivalent routines for the graphics statements. Notice that there is a separate position command for the text window. This only applies in the split screen mode.

If during the execution of a graphics command an error occurs, say the cursor is out of bounds, then the negative flag of the processor is set and the 'Y' register is set to the error code of the fault as described in the BASIC reference manual. The processor will exit the routine and, in my example, stop at the line called ERROR.

Assemble and execute the program using either the Macro assembler or the Assembler/editor cartridge. You should see exactly the same picture as with the BASIC program. Modify a few of the coordinates to get a feel for the program then you may progress to more complex shapes, say an octagon!

You will notice that the print routine uses a table that holds pointers to the text and the length of the text. So to print a line of text which may be the 'nth' entry in the program set the 'Y' register to n x 4 and execute the PRINT routine. The graphic PLOT and DRAWTO commands can be adapted to be table driven. The programming possibilities are almost endless.

I hope you find this article useful and maybe we'll see some of your programs in future issues.

TEXTPRO MACROS
continued

first <=> it sees in the pair. Well, in fact, that is exactly what it is doing. So you would need to put a dummy character before the double <=>. [OPTION]_[+] the dummy character would actually type the second <=> in the editor and then, execute the first <=> macro command, so I would use a difficult-to-type dummy key like [ESC]_[SELECT]_[SHIFT]_[+] (inverse backslash).

This is just a theoretical discussion but it shows how the logic of macro execution operates.

NEXT TIME ...

In the next instalment I'll give you some exciting examples which bring together all we have learnt so far.

This article originally appeared in the U.S. magazine Current Notes which, alas, no longer caters for the Atari Classic.



PROGRAMMING

TWO MACHINE CODE MONITORS

Machine code programmers are always looking for ways to make their programming tasks easier. Here H.S. Wood presents a couple of utilities that could be just the job

When one intends to program in Machine Code or Assembly Language (the two are not the same) then a Monitor is essential. Such a Monitor allows assembled code to be examined and run. It will allow the program to be stopped at preset Break Points so that registers etc. can be examined and it will allow the program to be restarted as if nothing had happened. In other words a program can be "debugged".

The two programs on this issues's disk, ***MONITOR4.CMD*** and ***MONMAIN3.CMD***, carry out the above tasks. These Monitors are based upon the extended Monitor supplied in 1979 with the UK101 Kit computer but they are considerably more useful and are suitable for the ATARI 8 bit 800XL and 130XE.

DOUBLING UP

Why are there two almost identical programs? Well it is a question of location in memory. Both programs are 2K bytes long, but while MONITOR4 is located at $8800 to $9000, MONMAIN3 is under the Basic ROM at $A000 to A800. Each is independent of the other and can be used singly or both may be in memory at the same time.

If you want to examine the Basic ROM then you must use MONITOR4 but if you want to use locations $8800 to $9000 then you must use MONMAIN3.

MONITOR4.CMD, once loaded, can be accessed from Basic by the command G=USR(34816). To exit press RESET.

MONMAIN3.CMD cannot be accessed directly from Basic because it uses the Basic addresses. This is overcome by the use of a short routine (9 bytes) which is placed at $9C00 - the start of screen memory. From Basic G=USR(39936) will access MONMAIN3 and RESET will exit back to Basic. However, if you are using Graphics screens, then this 'access routine' will be overwritten. You overcome this by writing the 9 bytes somewhere else in memory (such as page 6). A short Basic routine can install these 9 bytes. Use MONMAIN3 to see the 9 bytes so that you can copy them.

Note also that MONMAIN3.CMD has a loading routine which switches off Basic when the program is Loaded from disk. This loading routine will overwrite some memory but is not used after loading is complete. To prevent problems MONMAIN3.CMD should be loaded at the start of a session if it is to be used.

HOW TO USE THE MONITORS

Both Monitors are identical in operation and Keys **A**, **B**, to **Z** control the functions. **SHIFT @** and **SHIFT 2** are also used.

When a key is pressed it must be followed by other input. Most keys expect an address (addr) in HEX but some keys need a number 1 to 8 - i.e. B1 E5 etc.

The CPU registers are displayed as you would expect with keys **A X Y P** and **K**.

P is the 'flags' register while **K** is the LSByte of the Stack Pointer (the MSByte is always $01).

Key **I** will display all CPU registers together.

Key **Q** *addr* will display one screen full of disassembled code starting at 'address'. Key **J** will show the next screenful, and key **Z** will show the next commands one at a time. **RETURN** will exit this function.

SHIFT @ *addr* will display the byte at 'address'. This byte may be changed by entering a new value and key **J** will go to the next address. RETURN will exit. **SHIFT @** can be used to enter Machine Code routines by hand if required. Be careful not to change Monitor locations.

One of the most important functions is to Run a program to find out why it does not work as you intended. The program must be in memory at a known address and keys **B C E G T** are used as follows:-

Decide where you would like the program to stop. Press **B** followed by a number **1** to **8** followed by the address. Press **T** and the addresses of the Break Points (BP's) will list out. Unused BP's will be $FFFF but the ones you have set will display their addresses.

If you are not satisfied with the BP's you have set press **E1** (or the number you entered) and that BP will be eliminated. Press Return to exit. Up to eight BP's can be set so that they will operate as the program reaches their addresses.

Next enter **G** *addr* (where 'address' is the start of the program) followed by Return. The program will stop at the BP and print out the address and all the CPU registers. The register values can be changed or other parts of the routine can be changed. When you are satisfied enter **C** to continue. In this way a program's progress can be followed through and any errors should be found.

NOTE to change a register, press its name (A etc.) and enter the new value. Press RETURN.

To calculate values in HEX press **H** as follows:-

H *number,number+*=answer - you enter the two values of 'number' and the + (which may be - * or /). The comma, the = and 'answer' are printed by the MONITOR.

To load a disk sector from drive 1 to Page 6 enter **L** sector where 'sector' is the sector number in HEX. Next enter **D 0600,0680** and the sector values will be Dumped on the screen. Alternatively **Q** may be used to provide an assembly listing. i.e. **Q 0600**.

As mentioned above entering **D** *addr,addr* will Dump the code from the first address to the second on to the screen. The screen will scroll so you might need to use CONTROL 1 to stop it.

Key **F** is a FILL. Press **F** *addr,addr*=byte (the Monitor inserts the comma and the equal). Memory from the first address to the second will be filled with byte. This is a good way to clear memory by using byte=00.

Keys **M** and **R** behave similarly. **M** copies a block of data while **R** relocates a copy of a block of data. The original data is unchanged unless it is overlapped by the block limits. The data Relocated will have its codes changed so that it will RUN at the new address. Move only moves the data.

R *addr=addr1,addr2* will Relocate from 'address' to a block starting at address1 and ending at address2. The equal sign and the comma are put in the line by the Monitor.

Keys **N** and **W** behave similarly. **N** searches for a sequence of bytes (from one to eight) while W searches for a string of ASCII characters (1 to 8). The entries are:-

N *byte byte -- byte >addr,addr*

W *string >addr,addr*

The greater than sign '>' has to be entered when all the values have been put in but the MONITOR enters the comma. The MONITOR will search from the first address to the second for a match. When a match is found, the address of the first character and its value is put on the screen where it can be changed if required. Using key **J** steps through the memory. Return will return to MONITOR.

SHIFT 2 will display the ASCII value of the character for the value shown when using SHIFT @ or the N and W functions.

SPARE KEYS

WARNING although addresses are given below for MONMAIN3.CMD it is recommended that if these are used it is only on a temporary basis. This is because MONMAIN3.CMD has a loading routine and if it is re-saved it will not load again. In any case it is important that only copies of these MONITORs are modified.

Keys O S U and V are not used by MONITOR and can be set to point to a User routine if required. The addresses are as follows:-

KEY	MON4	MON3
O	$897E/F	$A17E/F
S	$8986/7	$A186/7
U	$898A/B	$A18A/B
V	$898C/D	$A18C/D

To put an address in the above locations (say Key O) press **SHIFT @ 897E.** Enter the LSByte of the address. Press **J** and enter the MSByte. Press Return. Check using **Q 897E.** Future presses of key O will run the routine at the entered address. If you put 9F (LSByte) and 17 (MSByte) then KEY O will go to DOS.

If the change is to be permanent (MONITOR4.CMD only) then the MONITOR must be saved using DOS. Use a different number to identify the changed program.

The two programs mentioned in this article are on this issue's disk ready to run. They are not available as a type-in listing. •

THE ACCESSORY SHOP

DISKS ... DISKS ...

NIBBLER
Slinking through an unknown land Nibbler finds himself in an iridescent world of constant change. As time grows short his body grows longer and he risks running into himself. Can he survive?

CLOSEOUT PRICE 50p plus 50p p&p

RAMPAGE
It's rough and tumble all the way as you control three incredibly nasty characters which bear a remarkable resemblance to King Kong, Godzilla and Wolf-man through an orgy of destruction in Chicago, New York and San Francisco. You have 150 days of destruction in 50 different cities. Time for some revenge!

CLOSEOUT PRICE £1.00 plus 70p p&p

DRUID
Another classic in which, as Last of the Great Druids, you wander through the Dungeons of Acamantor with your mighty Golem by your side on a quest to destroy the four demon princes. As you delve deeper into the dungeons you will find chests containing spells of formidable power to aid you and Pentograms of Life which will heal and revitalise you. Excellent graphics and superb gameplay make this one of the best arcade games.

CLOSEOUT PRICE £1.00 plus 70p p&p

MAXWELL'S DEMON
A simple yet challenging puzzle game which will get you totally addicted. Two games in one for double the fun

CLOSEOUT PRICE 50p plus 50p p&p

CASSETTES

LANCELOT
Level 9 recreate the time of wizards and the Knights of the Round Table in a three part adventure spanning the complete saga of the quest for the Holy Grail. Superbly researched and superbly written this text adventure is full of atmosphere and highly recommended.

CLOSEOUT PRICE £1.00 plus 90p p&p

JUGGLE'S HOUSE
Another neat program for 3 to 6 year olds teachin g the concepts of inside, outside, upper and lower with Juggle's house and toyshop

CLOSEOUT PRICE £1.00 plus 70p p&p

BATTALION COMMANDER
An exciting real-time tactical game with you as lieutenant colonel in charge of an entire armoured battalion. Choose from five different scenarios from a training mission against a Soviet tank battalion to tough assignments against the Chinese. You can adjust the relative strengths of your forces and the opposition and choose from 40 different maps and 5 different scenarios.

CLOSEOUT PRICE £1.00 plus 70p p&p

LAST FEW ROMS

POLE POSITION
ONLY 8 LEFT!

CLOSEOUT PRICE £1.50 plus 70p p&p

BASIC CARTRIDGE
Some programs, especially some early public domain programs won't run on the XL/XE but they will if you plug in the old version of Basic.

CLOSEOUT PRICE £1.00 plus 70p p&p

DESIGNERS PENCIL
A new programing language that is based around the creation of graphics screens. Difficult to explain but worth a go at this price!

CLOSEOUT PRICE £1.00 plus 70p p&p

Transfer these cassettes to disk!



TRANSDISK IV
is still available

Tells you how to transfer the software shown opposite - but you can do many more - over 300 programs successfully transferred!

TRANSDISK IS NOW ONLY £9.95
(previously £14.95)

AVAIALABLE ONLY FROM PAGE 6

BARGAIN CASSETTES

Your choice of
any 5 cassettes for £1.50 plus 80p p&p
any 10 cassettes for £2.00 plus £1.20 p&p

180 ✓
Eight opponents, digitised speech, two player option. live joystick control, full matchplay scoring, superb playability

BOMB FUSION ✓
A terrorist gang has planted bombs in the Nuclear Processing Plant and you have to go in to save the plant

DESPATCH RIDER ✓
Join the growing band of street demons who terrify the population of the big cities to get the parcels through.

FEUD ✓
You must "out-spell" your rival Wizard. Tread carefully as strange things can happen in this game of magic

FOOTBALL MANAGER
Everything you might want in a managerial simulation. Far too many features to describe, but you won't be disappointed

GHOSTBUSTERS ✓
At this very moment hundreds of ghosts are making their way to the infamous spook central. Only you can save the world from disaster

GUN LAW ✓
Four months of bloody alien attacks have taken their toll. You are left to fight alone against ruthless and bloodthirsty killers with just a single machine gun

HENRY'S HOUSE ✓
Little Henry has shrunk and must navigate his way through the royal household to find the cure. Voted one of the all time great games

INVASION ✓
Mobilize your units and prepare for battle. This all action space conflict requires skill, strategy and tactics.

KIKSTART ✓
The ultimate off-road motorbike scramble. Guide your rider over the obstacles in this great game for 1 or 2 players

LOS ANGELES SWAT ✓
Rescue the hostages from the terrorist gangs holding out in West L.A. Clean up the streets. Blow away the bad guys

MASTER CHESS ✓
An excellent chess simulation with all the correct moves, various openings, in fact everything to keep you challenged

MILK RACE ✓
Cycling 1,000 miles is no mean feat - and you could end up feeling pretty exhausted by the time you've finished

MR DIG ✓
An old favourite in which Mr Dig has to dig for hidden food supplies in the 'Meanie' territory below ground.

NINJA ✓
Blasts the belt off all other martial arts games! It says on the inlay! Someone sure reckons this is the best punching, kicking, ducking and diving game of all

ON CUE ✓
A challenging real life simulation which combines Pool and Snooker on the same cassette. An absolute must for both enthusiasts and beginners alike.

PANTHER ✓
Save the last humans on Xenon. Take your ground attack ship through this 3D scrolling mega shoot-em-up with great graphics and unbelievable soundtrack

PENGON ✓
Can you save Penguin Willy from the ferocious mutant sea lions? Stun them by knocking them against the walls or crush them with sliding ice blocks

PLASTRON ✓
Take your place in a small band of pirates out to steal fossil fuels from the biggest corporation in the galaxy.

PROTECTOR ✓
Assigned to the US Army Helicopter Training School your aim is to become the best chopper pilot in the West

REVENGE II ✓
The Mutated 90 foot high, laser spitting death camels have rebelled against their captors the Zzyaxians and are out for revenge! A Jeff Minter classic

ROCKFORD ✓
The only true arcade version of the classic game Boulderdash. Explore 4 levels on each of 5 different worlds

SIDEWINDER II
It is 27 years since the final battle of the war with the aliens. All this is about to change. Step aboard your craft to defend mankind in this space blast

SPEED HAWK
Defend the ring worlds of your solar system from space pirates. Another of the great space games

SPEED ZONE ✓
Enter the Speedzone in a frantic defence against alien forces. A survey ship comes under attack and your "Starfire" class attack craft is launched

STAR RAIDERS ✓
What more can be said. Probably the best computer game in the world - ever!

TAIL OF BETA LYRAE ✓
Our A-Z of Atari Software series says "The ultimate 'Scramble' clone with superb graphics and music."

TWILIGHT WORLD ✓
Equipped with the latest in anti-gravity pods and Laser weaponry, battle your way through each of eleven dungeons

UNIVERSAL HERO ✓
Only seven seconds left to save the planet! Universal Hero has to save his skin and everybody else's by finding bits to repair a shuttle

✓ TRANSDISK IV shows you how to transfer these to disk!

ORDER ITEMS FROM THE ACCESSORY SHOP WITH THE ORDER FORM ENCLOSED WITH THIS ISSUE OR WRITE TO
PAGE 6, P.O. BOX 54, STAFFORD, ST16 1DR
TELEPHONE ORDERS ACCEPTED ON 01785 241153 USING ACCESS OR VISA



PROGRAMMING

MULTI-POKE

Atari Basic is one of the best Basics of all but there are areas that could be improved. John Foskett has found a way to add a simple command that can save a great deal of time

There are times when writing even a simple program where it is necessary to calculate the high/low bytes of an address such as when using custom display lists, interrupts, etc. and Turbo BASIC has two commands which simplify this problem. Anybody who programs in Turbo BASIC will be aware of its double poke (DPOKE) and double peek (DPEEK) commands which eases this process by accessing the stated location and the following location together at the same time, doing the calculations for you automatically.

As an example, consider the following two comparisons, firstly when finding the address of the display list ....

Using Atari BASIC....

```
100 DL=PEEK(560)+256*PEEK(561)
```

But in Turbo BASIC, it is simply ....

```
100 DL=DPEEK(560)
```

Secondly when finding the address of, for example, a display list interrupt string (DLI$) and setting the vector to point it ....

Using Atari BASIC the routine is....

```
100 I=ADR(DLI$)
110 HI=INT(I/256)
120 LO=I-HI*256
130 POKE 512,LO
140 POKE 513,HI
```

But in Turbo BASIC, it is simply ....

```
100 DPOKE 512,ADR(DLI$)
```

The routine presented here performs both the DPOKE and DPEEK functions for Atari BASIC, automatically selecting which is required. The routine also performs a normal single POKE as well, the DPOKE function again automatically determining which is required. Also the routine is multi-function and will perform as many POKEs and DPOKEs in any combination as required in one operation.

USING THE ROUTINE

The function of the routine is determined by the number of parameters used in the call. When used with only one parameter, an address, the routine performs the DPEEK function returning the combined contents of the specified location and the location which follows to BASIC. When used with any number of pairs of parameters, the routine performs the multi-POKE/DPOKE function by POKEing (or DPOKEing) the address parameters which is the first of each pair with the value of the parameters which follow, that is the second parameter of each pair. The routine will perform a minimum of one POKE or DPOKE function using two parameters, an address and a value and the maximum is limited by the number of parameters that can be included in the call. Apart from using a single parameter for the DPEEK function, the routine must be used with an even number of parameters otherwise a crash may occur.

To clarify how the routine is used for DPEEK (double PEEK).

In this example, for finding the address of the display list ....

```
DLIST=USR(ADR(DPK$),560)
```

(Note the use of ONE parameter)

The equivalent in Turbo BASIC is ....

```
DLIST=DPEEK(560)
```

or the equivalent in Atari BASIC....

```
DLIST=PEEK(560)+256*PEEK(561)
```

To clarify how the routine is used for DPOKE (double POKE)....

This example demonstrates using the routine to set the display list address vector to point to page 6 (1536)....

```
I=USR(ADR(DPK$),560,1536)
```

(Note the use of TWO parameters)

The equivalent in Turbo BASIC is....

```
DPOKE 560,1536
```

```
; MULTI-POKE/DPOKE/DPEEK
; for Atari BASIC
; Version I Jan'96
; Written by John Foskett
;-----------------------
;
; Define Labels
ADDLO = 203
ADDHI = 204
VALHI = 205
;
; Start of Code
 PLA
 CMP #1
 BEQ DPEEK
; POKE/DPOKE
 TAX
LOOP
 LDY #0
 PLA
 STA ADDHI
 PLA
 STA ADDLO
 PLA
 STA VALHI
 PLA
 STA (ADDLO),Y
 LDA VALHI
 BEQ SINGLE
 INY
 STA (ADDLO),Y
SINGLE
 DEX
 DEX
 BNE LOOP
 RTS
; DPEEK
DPEEK
 PLA
 STA ADDHI
 PLA
 STA ADDLO
 LDY #0
 LDA (ADDLO),Y
 STA 212
 INY
 LDA (ADDLO),Y
 STA 213
 RTS
```

```
HY 10 REM ********************************
OT 20 REM *  MULTI-POKE/DPOKE/DPEEK   *
IM 30 REM *  A MACHINE CODE ROUTINE   *
QY 40 REM * FOR USE WITH ATARI BASIC  *
CU 50 REM * WITH DEMONSTRATION PROGRAM *
CV 60 REM *---------------------------*
CC 70 REM * BY JOHN FOSKETT   JAN'96  *
RE 80 REM *     FOR NEW ATARI USER    *
IG 90 REM ********************************
UR 100 DIM DPK$(50)
AQ 110 DPK$="h1[A]p[ESC,ESCAPE]* [,]h<E>
   Lh<E>Kh<E>Mh<Q>K/Mp[C]H<Q>KJJPq
[.]h<E>Lh<E>K [,]1K<E>TH1K<E>U[.]
   "
QQ 120 REM
QS 130 REM
FU 140 REM  DEMONSTRATION PROGRAM
RA 150 GRAPHICS 0:POKE 16,64:POKE 53774,6
   4
QF 160 RESTORE 360:FOR I=0 TO 15:READ J:P
   OKE 1536+I,J:NEXT I
JT 170 REM  Get Display list DPEEK(560)
KP 180 DLIST=USR(ADR(DPK$),560)
NU 190 REM  Multi-POKE/DPOKE
FL 200 I=USR(ADR(DPK$),DLIST+27,130,512,1
   536,54286,192,752,1)
AE 210 REM  Find & DPOKE Random Number
MW 220 R=INT(RND(0)*65280)+256:I=USR(ADR(
   DPK$),709,R)
LG 230 POSITION 1,0:? "MULTI-POKE/DPOKE/D
   PEEK FOR ATARI BASIC":? "[ESC,TAB]
   [ESC,TAB]WRITTEN BY":? "[ESC,TAB]
   [ESC,TAB] [ESC,LEFT]JOHN FOSKETT"
MD 240 ? :? " SAME AS TURBOS  DPOKE  AND
    DPEEK    BUT WITH MULTI-FUNCTION FAC
   ILITIES"
VM 250 ? :? "In this demonstration progra
   m, the 2  colour registers 709 & 710 a
   re being"
OB 260 ? "DPOKEd with a random number bet
   ween   256 and 65535 (inclusive) using
   ...."
AP 270 ? :? "[ESC,TAB]I=USR(ADR(DPK$),709
   ,";R;")":? :? "As a result...."
XY 280 ? "PEEK(709)=";PEEK(709);" and PEE
   K(710)=";PEEK(710)
UA 290 ? :? "Using this routine for DPEEK
   ....":? "  PRINT USR(ADR(DPK$),709) =
   ";
HA 300 REM  Print DPEEK(709)
YW 310 ? USR(ADR(DPK$),709)
NF 320 ? :? "The machine code routine is
   also used in setting up this demo prog
   ram. See"
CT 330 ? "lines 180, 200, 220 and 310.":?
    "[ESC,TAB]PRESS  START  TO RUN AGAIN"
   ;
BA 340 ON PEEK(53279)<>6 GOTO 340:RUN
UA 350 REM  DLI Data
MP 360 DATA 72,169,10,141,10,212,141,23,2
   08,169,148,141,24,208,104,64
```

Underline = INVERSE CHARACTERS · [ ] = CONTROL + CHARACTER · < > = INVERSE CONTROL + CHARACTER

or the equivalent in Atari BASIC....

POKE 560,0:POKE 561,6, but if the address was that of a string, then the Lo/Hi bytes would have to be calculated first as previously described.

To clarify how the routine is used for multiple POKE/DPOKE functions ....

I=USR(ADR(DPK$),54286,0,512,ADR(DLI$),548,ADR(VBI$),54286,192,560,1536,82,0,752,1)

The above shows a typical use for the routine where several POKEs and DPOKEs are performed at the same time. The equivalent (in Turbo BASIC) is....

POKE 54286,0
DPOKE 512,ADR(DLI$) (DLI as a string)
DPOKE 548,ADR(VBI$) (VBI as a string)
POKE 54286,192
DPOKE 560,1536


```
UA 10 REM ****************************
OK 20 REM *  MULTI-POKE/DPOKE/DPEEK  *
AI 30 REM *   MACHINE CODE ROUTINE   *
UQ 40 REM * CHARACTER STRING WRITER  *
YL 50 REM * BY JOHN FOSKETT   FEB'96 *
UF 60 REM ****************************
FI 70 GRAPHICS 0:POKE 16,64:POKE 53774,64
   :DIM F$(14):F$="D:STRING.LST"
KF 80 POKE 752,1:? :? "CHARACTER STRING W
   RITER":? :? "PRESS START TO BEGIN"
PT 90 ON PEEK(53279)<>6 GOTO 90:? :? "WRI
   TING TO DISK - PLEASE WAIT":RESTORE :O
   PEN #1,8,0,F$
WM 100 REM  Write MC String
QD 110 ? #1;"110DPK$=";CHR$(34);
AM 120 READ J:IF J<0 THEN 150
XY 130 ? #1;CHR$(J);:GOTO 120
BS 140 REM  End
GZ 150 ? #1;CHR$(34):CLOSE #1:? :? "FILE
   WRITTEN IN THE LIST FORMAT":? :? "FILE
    NAME: ";F$(3);"[ESC,BELL]":END
EE 160 REM  Machine Code String Data
JC 170 DATA 104,201,1,240,27,170,160,0,10
   4,133,204,104,133,203,104,133,205,104,
   145,203,165,205,240,3,200,145,203,202
PL 180 DATA 202,208,231,96,104,133,204,10
   4,133,203,160,0,177,203,133,212,200,17
   7,203,133,213,96,-1
```

Underline = INVERSE CHARACTERS · [ ] = CONTROL + CHARACTER · < > = INVERSE CONTROL + CHARACTER

POKE 82,0
POKE 752,1

The above shows a typical initialising procedure for setting up a custom display list, a VBI and a DLI. It also shows the left margin reset to zero and the cursor being turned off.

Another advantage of the above is in program protection. If you were to use variables in the place of some of the constants and then change them for protection in the normal way, nobody would know what this weird looking USR statement was in your program even if they did manage to break into it!

HOW THE ROUTINE WORKS

The top part of the source code listing performs the POKE and DPOKE functions whilst the bottom part of the listing performs the DPEEK function. The first action of the routine is to remove the first parameter from the stack and use it to determine whether a POKE/DPOKE or a DPEEK function is required by comparing it with the number one. If the value was one which means that only one parameter was used in the command, then control is passed to the DPEEK routine at the bottom of the listing otherwise a POKE or DPOKE function is assumed.

Both sections are described separately as follows ....

THE POKE/DPOKE SECTION

This section basically forms a loop which actions the parameters in pairs, that is an address followed by a value. Prior to entering the loop, the number of parameters used in the command (from the above mentioned comparison) is transferred into the X register using TAX. The loop first removes a pair of parameters from the stack, actions them and then decrements the X register using DEX twice (that is once for each parameter). It then follows that the X register contains a record of the number of parameters that remain on the stack at all times such that when the X register is zero which means that there are no more parameters on the stack, the routine is exited via RTS otherwise the loop is repeated.

This forms a very flexible routine which enables any number of parameters to be used in the command so long as they are in pairs.

In the loop the two parameters removed from the stack are stored in page zero addresses. The first parameter (the address) is stored in locations ADDLO and ADDHI in the normal 2 byte lo/hi format which are defined at the top of the listing as locations 203 and 204. Any of the unused locations may be used for this purpose but they must be in page zero. The high byte of the second parameter (the value) is stored in location VALHI defined as location 205 at the top of the listing. Any unused location anywhere in RAM may be used for this purpose. There is no temporary storage for the low byte of the value parameter because this is stored directly into the specified address using indirect indexed addressing. The accumulator is then loaded with the high byte of the value from VALHI which is then checked for zero using BEQ and if found, access to the second memory location (specified address plus one) is bypassed so that the routine actions a single POKE. If the contents of VALHI was not zero, then the original value of the parameter was greater than 255 and hence a double DPOKE is actioned instead, incrementing the Y register to enable access to the next location again via indirect indexed addressing.

Note that if an ordinary single POKE is assumed because the value parameter is less than 256, then there is no access to the next address in sequence such that its contents remain unchanged. This is the only difference between the routine and Turbo BASICs DPOKE command where DPOKE actually does access the next memory location in sequence, loading it with a zero if the value is less the 256. Allowing the use of single POKEs in this way was deemed to be more important because it allows the advantage of multi-function, that is of combined POKEs and DPOKEs.

THE DPEEK SECTION

As previously stated, control is passed to this section whenever there is only one parameter used in the USR command. This section first removes the parameter (the address for DPEEKing) from the stack and stores it in the previously mentioned locations ADDLO and ADDHI. Next the Y register is zeroed to allow access to the specified address via indirect indexed addressing the contents of which are stored in location 212, the low byte of the value to be returned to BASIC. The Y register is then incremented to allow access to the next location in sequence again via indirect indexed addressing and its contents are stored in location 213, high byte of the value to be returned to BASIC. The final value returned to BASIC is automatically calculated from the lo/hi bytes stored in locations 212 and 213 by the floating point routines.

THE DEMONSTRATION PROGRAM

The actual machine code routine is the string immediately below the REM header and the rest of the listing is the demonstration program.

The demonstration program shows how the routine is used by DPOKEing the two screen colour locations 709 and 710 with a random number between 256 and 65535. As a result, the screen is coloured accordingly and the contents of the two individual locations are displayed on screen separately along with their combined value found by DPEEK. Note that the value found is always the same as the original random number.

Please bear in mind that because the screen is randomly coloured, the brilliance of the characters may equal that of the screen itself resulting in a blank screen. The program includes a small DLI routine to ensure that the bottom line of the screen is always visible for displaying the PRESS START prompt.

Note that the demonstration program uses this routine itself as follows ....

Line 180 gets the address of the display list by DPEEKing thus ....

```
DLIST=USR(ADR(DPK$),560)
```

Line 200 performs multi-POKE/DPOKE functions using ....

```
I=USR(ADR(DPK$),DLIST+27,130,512,
1536,54286,192,752,1)
```

This represents ....

```
POKE DLIST+27,130
DPOKE 512,1536
POKE 54286,192
POKE 752,1
```

Line 220 DPOKEs the random number into locations 709/710 using ....

```
I=USR(ADR(DPK$),709,R)
```

Line 310 prints the value found by DPEEKing location 709 using ....

```
PRINT USR(ADR(DPK$),709)
```

TYPING THE MACHINE CODE STRING

Because of the difficulty involved with typing machine code strings, the second program has been included to write the string for you from a bank of numeric data. Simply run the program and press START after which the string will be written to disk using the file name "STRING.LST" in the LIST format. It is then a simple matter to ENTER the string from disk. ●



REVIEWS

VAGABOND

reviewed by Paul Rixon

A new game? Well no, but it's a welcome addition to Micro Discount's unique range of imported titles. In common with many others, Vagabond (or "Wloczykij" for those with linguistic talent) originates from Poland. The game was published by LK Avalon in 1993 and, judging by the shape of its glossy insert, was probably intended for circulation on tape.

Thanks to the efforts of MD's interpreter, an English version of the story sheet is provided. We're told that Vagabond - an evidently cheerful green 'blob' with blue arms, blue legs and a red nose - has returned home from a characteristically idle wander and is keen to get some rest. But his lackadaisical intentions are thwarted when a friendly alien knocks on the front door (as they do?) and asks for help in retrieving the passengers of a stricken space ship. Hmm!

This means, of course, that your job is to help Vagabond jump onto platforms, collect passengers and steer clear of obligatory nasties. The lower two-thirds of the playfield encompass the side-on view of a colourful landscape. This incorporates platforms, clouds, trees, gates and similar forms of scenery which slide horizontally as your character strolls around. The upper screen portion displays counters to represent the number of lives and passengers remaining. There's also an egg-timer so you won't be tempted to loiter.

The joystick controls are straightforward. Pushing left or right prompts Vagabond to walk in the corresponding direction, whilst an upwards push makes him jump - in a crazy, head-over-heels fashion. As he leaps, you can make horizontal adjustments to improve the chances of a safe landing. Some nasties are static and must be avoided, but others are mobile and can be knocked out by a well-timed manoeuvre. There are no bonus points for aggressive behaviour, so it's probably wise to concentrate on the key task of rescuing helpless passengers. A specified quota must be found before you can access the following level.

Vagabond benefits from a high standard of visual design. Music is also good, though fairly repetitive, and there's an alternative option of sound effects. Playability wise, the first level is more demanding than some of the later ones - persevere and your confidence grows. The game could do with a highscore table or a level-password access system, but otherwise it must be recommended.

Title:	**WLOCZYKIJ (VAGABOND)**
Publisher:	**LK Avalon**
Supplier:	**Micro Discount**
Format:	**Disk**
Price:	**£4.00**

contact ... contact ... contact ...

FOR SALE

FREE/SWAP: To a good home - cassette games 30+. Free to anyone in South Lincs/North Cambs area else pay postage. Jason Kendall, 22 Beck Way, Thurlby, Lincs PE10 0LE. Tel. 01778 426409

MAGAZINES: I have all the issues of ATARI USER magazine (Vol. 1/No. 1 May 1985 to Vol. 4/No.7 November 1988) in excellent condition. I am willing to sell them for any good price offered. If you are interested please let me know by mail or fax. Lappas Nestor, P.O. Box 8556, GR-100 10 ATHENS, Greece. FAX No. 0030 1 4923236

SOFTWARE/HARDWARE: Atari 8-bit software & hardware - tape/disk/ROM software; computer/printer/tape decks/disk drives. Send SAE for list and prices to Dan Baverstock, 199 Mackie Avenue, Patcham, Brighton, East Sussex BN1 8SE or call 01273 883630. Reasonable prices for individual items or bulk buys. Will consider swapping.

XL SYSTEM: Atari 800XL, 1050 disk drive, 1010 tape deck, 1027 printer. All with manuals and in working order with leads. Offers to D. Neal on 0117 9564842

HARDWARE: 65XE computer, 800XL computer, 1029 printer, 1027 LW printer, 1020 Printer/Plotter, TWO 1050 disk drives, XC12 Datacorder, 3 joysticks. All in perfect working order and boxed with leads and manuals plus games on tape/disk and loads of utilities. £400 the lot. For details tel. Geoff on 01670 364068 after 7pm

WANTED

SOFTWARE WANTED: Disk, ROM or tape: Fruit Picking, Winter Olympiad '88, Rally Speedway, Alien Blast, Crossbow, A Day at The Races, any old APX stuff. Jason Kendall, 22 Beck Way, Thurlby, Lincs PE10 0LE. Tel. 01778

SPEECH CHIP: Wanted Speech Synthesiser Chip No. SP0256. Phone George on 01606-76062

SLINGSHOT: Original copy of Slingshot and Con Kit on tape or disk or Transdisked copy. Please write to Mr P Cole, 29 Frittenden Close, Stanhope, Ashford, Kent TN23 5SX

THE TRUTH IS OUT THERE: Or maybe someone with a Supra Hard Drive and Interface unit to plug into a 130XE. Or an interface to plug a third party Hard Drive into the 130XE or any technical help to achieve the above match. So make my day and contact Paul Kovacs, 17 Ingleby Moor Crescent, Firthmoor, Darlington. Co. Durham DL1 4RW

FREE TO SUBSCRIBERS

The CONTACT column is free of charge to subscribers who wish to sell their equipment or contact other readers. Space is limited so we request that entries be kept as short as possible. Extremely long entries may be heavily edited or ignored. Send your CONTACT notice on a separate sheet of paper (not as part of a letter) to:
CONTACT, PAGE 6 PUBLISHING, STAFFORD, ST16 1DR

FOR SALE ... WANTED ... PEN PALS ... ADVICE ... HELP

PROGRAM LISTINGS

Certain program listings which are too long to include in the magazine may be obtained free of charge as printed listings to type in. All programs are, however, included on the Issue Disk which is available with each issue. Remember this disk also includes BONUS PROGRAMS which do not appear in the magazine. If you would like the type-in listings please write or telephone indicating which listings you require. Please note that there are not necessarily extra listings for every magazine.

Write to LISTINGS, NEW ATARI USER, P.O. BOX 54, STAFFORD, ST16 1DR or telephone 01785 241153